I&EC research
Industrial & Engineering Chemistry Research

Article

Cite This: Ind. Eng. Chem. Res. 2019, 58, 11521−11531

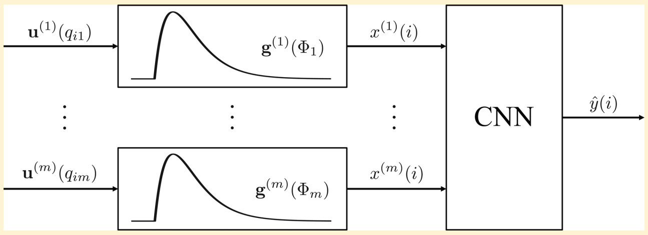
pubs.acs.org/IECR

Dynamic Soft Sensor Development Based on Convolutional Neural Networks

Kangcheng Wang,[†,‡] Chao Shang,[†,‡] Lei Liu,[†,‡] Yongheng Jiang,[†,‡] Dexian Huang,[†,‡] and Fan Yang*[,†,‡]

[†]Beijing National Research Center for Information Science and Technology (BNRist), Beijing 100084, People's Republic of China
[‡]Department of Automation, Tsinghua University, Beijing 100084, People's Republic of China

ABSTRACT: In industrial processes, soft sensor models are commonly developed to estimate values of quality-relevant variables in real time. In order to take advantage of the correlations between process variables, two convolutional neural network (CNN)-based soft sensor models are developed in this work. By making use of the unique architecture of CNN, the first model is capable of utilizing abundant process data, and the complexity of this model remains low. The second model integrates finite impulse response and CNN, and process dynamics can be reasonably embraced in this model. The effectiveness of the two models is validated by a simulation case and a chemical industrial case. Since the finite impulse response−convolutional neural network (FIR-CNN) can give the best prediction accuracy and the most interpretable trend of a quality-relevant variable, it has promising application potential in the chemical industry.


$\mathbf{u}^{(1)}(q_{i1})$
$\mathbf{g}^{(1)}(\Phi_1)$
$x^{(1)}(i)$
$\mathbf{u}^{(m)}(q_{im})$
$\mathbf{g}^{(m)}(\Phi_m)$
$x^{(m)}(i)$
CNN
$\hat{y}(i)$

1. INTRODUCTION

In order to improve product quality in industrial processes, one of the most important tasks is to measure some essential quality-relevant variables in real time. However, those variables may be difficult to measure online, because of unacceptable measurement time delay, or exorbitant costs on analytical instruments, off-line analysis, and device maintenance.[1] Motivated by the main idea of inferential control,[2] soft sensor models are derived. The function of a soft sensor is to make the online estimation of those key quality variables by utilizing some other easy-to-measure process variables that are highly correlated to quality variables. The output variables of the soft sensor are also called primary variables, while the input variables are often called secondary variables.[3]

There are many broadly investigated soft sensor modeling methods such as principal component regression (PCR),[4] partial least-squares (PLS),[5,6] and artificial neural network (ANN).[7,8] Those models are called static soft sensor models because they have static model structures. Many industrial processes have strong dynamic features. In other words, the states of a process at the current snapshot is related to its former snapshots. In order to describe these features, many dynamic soft sensor models emerge. Those models can be classified into two categories by the way of utilizing historical process data.

The first category of models includes process dynamics by combining data of current and historical snapshots into a higher dimensional matrix. That is, augmenting process data sampled at the current snapshot with historical data. Some models augment historical data along the dimension of secondary variables. For instance, Ku et al.[9] proposed dynamic PCA (DPCA). Kaspar et al.[10] introduced dynamic partial least-squares (DPLS). Some other models that have a recurrent structure combine current and historical data into a three-dimensional (3D) matrix. For example, the recurrent neural network (RNN) was applied in the soft sensor field by Su et al.[11] Ke et al.[12] proposed a long short-term memory (LSTM)-based soft sensor. The main limitations of those models is when the time lag of the included historical data is large, the dimension of the model input will be very high. As a result, the complexity of those models will be quite high.

The second category of models contains process dynamics by considering process mechanisms. It can be known from the past experiences that an industrial process can be approximated by a series of second order plus dead time (SOPDT) models. Ma et al.[13] calculated the convolution of the impulse response of SOPDT models and historical process data, and the convolution results are served as the input of a PLS-based soft sensor model. As a result, the model structure is greatly simplified, and the model is more robust than many conventional models. Moreover, dynamic process information can be reasonably incorporated in the convolution results, even when the process and quality variables do not have the same sampling rate. Lu et al.[14] proposed an impulse response template (IRT)-based soft sensor modeling approach. In this

Special Issue: Sirish Shah Festschrift

Received: May 7, 2019
Revised: May 28, 2019
Accepted: May 28, 2019
Published: May 28, 2019

DOI: 10.1021/acs.iecr.9b02513

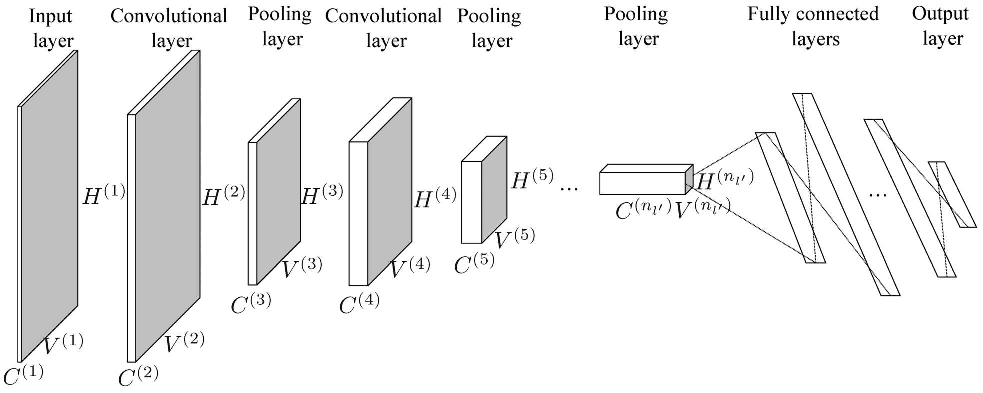


Figure 1. Typical architecture of CNN.

work, all of the parameters of impulse responses are constrained in a range given by mechanism analysis or practical experience, and those parameters are optimized by a modified differential evolution algorithm. In this way, overlearning of parameters can be alleviated, and the generalization ability of the model is greatly increased. Because IRT is a linear model and it has limited nonlinear representation ability, Shang et al.[15] proposed a soft sensor model with the Wiener structure. The first part of the model is a series of impulse responses, whose function is to describe the dynamic features of a process. The second part is a support vector machine (SVM) model, which is used to describe the nonlinear features. The parameters of both impulse responses and SVM are estimated in a Bayesian framework. Gao et al.[16] proposed a two-step alternate optimization method to train the parameters of impulse responses and SVM in the Wiener structure. Although the performance of IRT-SVM model is better than linear impulse response-based models, the nonlinear representation ability of SVM is still limited. Besides, the correlations between process variables are not utilized. As a result, the performance has a tendency to degrade when this category of soft sensor models are applied to real processes with strong nonlinearity.

In recent years, deep learning techniques have gained a growing influence in the field of machine learning.[17] Some models in the deep learning area are applied in the field of soft sensor development. Shang et al.[18] introduced a soft sensor model based on a deep belief network. By using abundant process data sampled between quality data to pretrain the network, satisfactory prediction results can be obtained. Yao et al.[19] proposed a hierarchical extreme learning-machine-based semisupervised deep learning soft sensor model. Ke et al.[12] proposed an LSTM-based soft sensor development approach. Yuan et al.[20] developed a variable-wise weighted deep stacked autoencoder-based soft sensor model. These models use various deep neural networks to extract features in industrial processes. However, the dynamic correlations that exist between process variables are rarely utilized in the existing deep-learning-based soft sensor models.

Recently, the convolutional neural network (CNN) plays an important role in deep learning, and it has been widely applied in numerous fields such as image recognition,[21] speech recognition,[22] and natural language processing.[23] Compared to other types of deep neural networks, the spatial relationship between adjacent pixels in visual input data can be perfectly utilized by the convolutional operation in CNN. Therefore, CNN outperforms many deep neural networks in many fields. CNN has also been applied to some industrial processes. Lee et al.[24] probed into the use of CNN in classifying bearing fault data. Lee et al.[25] introduced a novel CNN-based fault detection and classification model for semiconductor manufacturing. Zhu et al.[26] proposed a CNN-based soft sensor model to predict the next time point of key variables. Although these CNN-based models have some successful applications, there are two limitations that exist in these models. One is the correlations between process variables are not fully utilized in some models. The other is historical process data are equally treated, rather than given different weights according to their time snapshots. In this research, two types of CNN-based soft sensor models will be developed to overcome these limitations by considering process mechanisms and taking advantage of the CNN architecture.

The remainder of this article proceeds as follows. In section 2, the structure of CNN in image processing is first introduced, followed by the architecture of CNN. Then the modeling procedures of CNN-based soft sensor are reviewed. In section 3, FIR of SOPDT is revisited, and the modeling steps of FIR-CNN are described in detail. In order to demonstrate the effectiveness of the proposed models, a simulation case study and an industrial case study are employed in sections 4 and 5, respectively. Finally, several conclusions are summarized in the final section.

2. CNN-BASED SOFT SENSOR DEVELOPMENT

In this section, the architecture and basic knowledge of conventional CNN are first revisited. Then the CNN-based soft sensor is developed.

2.1. Conventional CNN.

The architecture of the original CNN proposed by Yan LeCun et al.[27] is illustrated in Figure 1. As shown in the sketch, there is a 3D tensor that has dimensions of $V^{(1)} \times H^{(1)} \times C^{(1)}$ in the input layer. There are several alternated convolutional layers and pooling layers behind the input layer. The output of the last pooling layer is connected to several fully connected layers, which are the same as the densely connected layers in a standard multilayer neural network. Assume that there are $n_{l'}$ convolutional and pooling layers, and the total number of layers in CNN is n_l.

Before building the CNN model, it is recommended to apply some preprocessing methods to the raw input data in advance. The z-score normalization is one of the most commonly applied preprocessing methods. The activation function in each layer is often chosen as a rectified linear unit (ReLU), sigmoid, or hyperbolic tangent, whose formulas are listed in eqs 1, 2, and 3, respectively.[28]

DOI: 10.1021/acs.iecr.9b02513

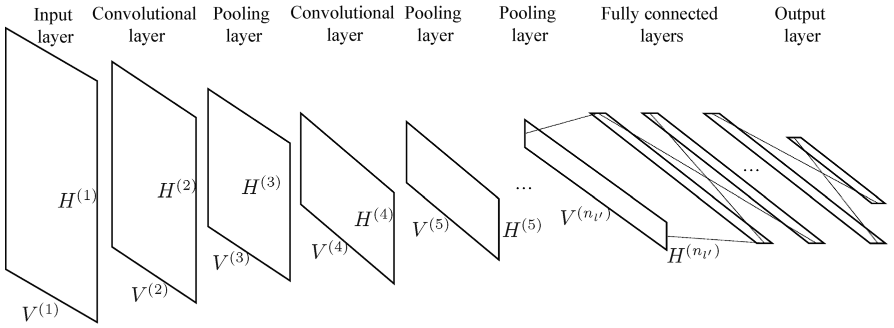


Figure 2. Architecture of the CNN-based soft sensor.

$$f(x) = \max(0, x) \tag{1}$$

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

$$f(x) = \frac{e^{x} + e^{-x}}{e^{x} + e^{-x}} \tag{3}$$

In CNN, the spatial relationships between adjacent pixels can be perfectly utilized by convolution operations. By sharing parameters and pooling, the network complexity of CNN is dramatically reduced. Moreover, the nonlinear representation ability of CNN can be enforced by nonlinear activation functions. Therefore, the performance of CNN is much better than many conventional classification or regression models.

2.2. CNN-Based Soft Sensor Development. In a stable closed-loop industrial process, high dynamic correlations exist between process variables. CNN is capable of utilizing the correlations between adjacent elements of the input tensor. Therefore, correlations between process variables can be used by CNN. Moreover, industrial processes often have evident dynamic features. In order to utilize these features to improve the performance of soft sensor models, a large portion of historical process data is often included in the model input. As a result, the model complexity will be rather large. CNN is capable of integrating a large amount of input data. By adjusting several parameters including kernel sizes, strides, and sizes of pooling windows, the complexity of CNN is greatly reduced. Therefore, CNN will be adopted in this research to build soft sensor models.

The architecture of CNN-based soft sensor is depicted in Figure 2. The input of CNN is process data. Similar to the CNN applied in computer vision, there are several convolutional and pooling layers, which are arranged alternately. The output of the last pooling layer is the input of fully connected layers, and the output of CNN is quality data.

Assume that there is one quality variable and N quality samples to be predicted, then the data of the primary variable can be expressed as $\mathbf{y}(i)$ $(i = 1, \cdots, N)$.

Assume that the number of secondary variables is m, and the jth secondary variable corresponding to the quality variable is denoted as $u^{(j)}(q_{ij})$ $(j = 1, ..., m,\ i = 1, ..., N)$. As the sampling interval of the quality variable is often much longer than that of process variables in many real industrial processes, there are a large number of process samples that exist between two quality samples. Considering the strong dynamics of industrial processes, not only $u^{(j)}(q_{ij})$ but also process samples before q_{ij}: $u^{(j)}(q_{ij} - k)$ $(k = 1, 2, ...)$ may be relevant to the corresponding quality sample $y(i)$. If the latest ds process samples are adopted, then the input matrix of CNN can be denoted as

$$\mathbf{U}(i) = [\mathbf{u}^{(1)}(q_{ij}), \mathbf{u}^{(2)}(q_{ij}), \cdots, \mathbf{u}^{(m)}(q_{ij})]^{\mathrm{T}} \in \mathbb{R}^{m \times d} \qquad i = 1, ..., N \tag{4}$$

where

$$\mathbf{u}^{(j)}(q_{ij}) = [u^{(j)}(q_{ij}), u^{(j)}(q_{ij} - 1), \cdots, u^{(j)}(q_{ij} - d + 1)]^{\mathrm{T}}$$
$$j = 1, ..., m,\ i = 1, ..., N \tag{5}$$

is the vector of the jth input variable, and d is the lag of historical input data.

Based on process mechanisms, some process variables will be chosen as secondary variables. Next, outliers in the selected process data need to be removed using the three-sigma rule. Then, z-score normalization should be performed to make all of the samples have zero means and unit variances. As process variables are correlated in a stable closed-loop industrial process, it is reasonable to calculate the convolution between kernels and all of the secondary variables. Therefore, in a convolutional layer, the width of convolution kernels are equal to the number of secondary variables m. Similarly, in a pooling layer, the width of the pooling window is assumed to be equal to the width of the input matrix, and the pooling window can only slide along the height of the input matrix.

The CNN-based soft sensor model is a regression model, whose optimization objective is to minimize the difference between the prediction values and the real values of the quality variable. Therefore, the cost function for the entire training set $\{(\mathbf{U}(1), \mathbf{y}(1)), ..., (\mathbf{U}(N), \mathbf{y}(N))\}$ can be defined as follows:

$$J(\mathbf{W}, \mathbf{b}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\|h_{\mathbf{W},\mathbf{b}}(\mathbf{U}(i)) - \mathbf{y}(i)\|^2)} \tag{6}$$

where $\mathbf{W}$ and $\mathbf{b}$ denote the weights and bias terms, respectively; and $h_{\mathbf{W},\mathbf{b}}(\mathbf{U}(i))$ denotes the output of CNN.

The optimizer of this optimization problem can be chosen as Adam, SGD, or RMSProp, which are frequently used in training CNN, RNN, and LSTM. Considering the stochastic property of the learning algorithm, the progress of model training is repeated several times and the model with the best performance on the validation data is chosen.

Note that, although deep learning models often have high complexity, the soft sensor models built in this article are all trained off-line. As a result, the time cost of online prediction of CNN is almost equal to that of an ANN. In addition, since there are only several hidden layers in CNN, and several essential hyper-parameters of CNN such as the number of convolutional layers, the number of kernels in each hidden layer, and the learning rate can be automatically determined by cross-validation, as described in ref 29, the parameter selection procedure is not too difficult.

The raw data is split into a training dataset, a validation dataset, and a test dataset. There are two main functions of the validation set, the first is to evaluate the model performance in the cross-validation procedure, and the hyper-parameters correspond to the least validation error is selected. The second is to avoid overfitting by terminating the learning progress when the validation error begins to increase.

3. FIR-CNN-BASED SOFT SENSOR DEVELOPMENT

Although CNN can utilize abundant historical process data in a simple manner, all of the historical data are equally weighted. This is short of the explicit process mechanism, because historical process data sampled at different times ought to have different impacts on quality data.

In order to solve this issue, a dynamic soft sensor with Wiener structure is proposed in this section. As shown in Figure 3, the soft sensor consists of two parts, the first is the finite impulse responses (FIRs) of SOPDT models, and the latter is a CNN. The outputs of FIRs are the input of CNN. Note that the reason for choosing the SOPDT model is that, although chemical processes often have complex dynamic properties, each submodel of a chemical process can be well-approximated by a SOPDT model. This is a rather common approach in industrial applications when the dynamic mechanism of the process is unknown. Note that only overdamped systems are considered in this paper.[14]

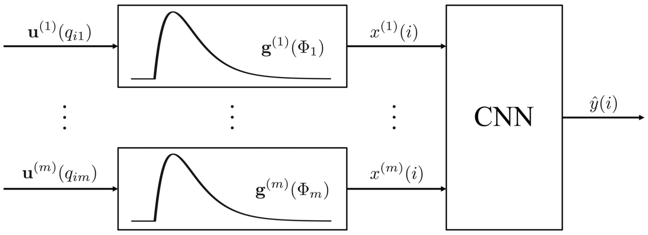

$\mathbf{u}^{(1)}(q_{i1})$
$\mathbf{g}^{(1)}(\Phi_1)$
$x^{(1)}(i)$
$\mathbf{u}^{(m)}(q_{im})$
$\mathbf{g}^{(m)}(\Phi_m)$
$x^{(m)}(i)$
CNN
$\hat{y}(i)$

Figure 3. Structure of FIR-CNN.

Although FIRs can reflect the intrinsic properties of chemical processes, they are linear models and lack nonlinear representation ability. Therefore, a nonlinear model should be followed behind the FIRs. From the previous section, we know that industrial process variables are correlated. As a result, the outputs of FIRs ought to be correlated. Compared to other nonlinear models, such as SVM, the correlations between intermediate variables can be embraced by CNN. Therefore, CNN is adopted as the nonlinear model. The proposed Wiener structure soft sensor is denoted as FIR-CNN. In FIR-CNN, each input $u^{(j)}(1 \le j \le m)$ corresponds to a secondary variable. The output of each FIR, denoted as $x^{(j)}(1 \le j \le m)$, is an intermediate variable.

3.1. FIR of SOPDT Revisit. In order to calculate the values of the intermediate variables, the FIR of the overdamped SOPDT model is briefly revisited. Assume that there are m secondary variables, and the relationship between the jth secondary variable ($j = 1, 2, ..., m$) and the jth intermediate variable can be depicted by the transfer function of an overdamped SOPDT model, which can be expressed as

$$G^{(j)}(s) = \frac{K_j}{(T_{1j}s + 1)(T_{2j}s + 1)} e^{-\tau_j s} \tag{7}$$

where K_j denotes the gain of the jth SOPDT model. T_{1j} and T_{2j} denote the time constants, and τ_j denotes the time delay. Assume that T_{sj} denotes the sampling interval of the jth secondary variable, and generally $T_{sj} \ll \min(T_{1j}, T_{2j})$. The infinite impulse response (IIR) of the jth SOPDT model can be derived as follows:

$$g^{(j)}(t) = \begin{cases} 0 & 0 < t < \dfrac{\tau_j}{T_{sj}} \\ \dfrac{K_j}{T_{1j} - T_{2j}}[\mathrm{e}^{-\left(t-\tau_j/T_{sj}\right)/\left(T_{1j}/T_{sj}\right)} \\ \quad - \mathrm{e}^{-\left(t-\tau_j/T_{sj}\right)/(T_{2j}/T_{sj})}] & t \ge \dfrac{\tau_j}{T_{sj}} \end{cases} \tag{8}$$

Given inputs $u^{(j)}(t)(j = 1, 2, ..., m)$, then the outputs can be expressed as

$$o^{(j)}(t) = \sum_{k=1}^{t} g^{(j)}(k)u^{(j)}(t + 1 - k)T_{sj} \tag{9}$$

Then the ith sample of the jth intermediate variable can be derived as

$$x^{(j)}(i) = o^{j}(q_{ij}) = \sum_{k=1}^{q_{ij}} g^{(j)}(k)u^{(j)}(q_{ij} + 1 - k)T_{sj} \tag{10}$$

Because the value of IIR is close to zero when the sampling time is larger than T_{lj}, where T_{lj} denotes the settling time, FIR can be an alternative to IIR. By selecting the first L ($L \ge \mathrm{round}(T_{lj}/T_{sj})$) values of IIR, the jth intermediate variable can be formulated as

$$\begin{aligned} x^{(j)}(i) &= \sum_{k=1}^{L} g^{(j)}(k)u^{(j)}(q_{ij} + 1 - k)T_{sj} \\ &= \mathbf{g}^{(j)\mathrm{T}}(\Phi_j)\mathbf{u}^{(j)}(q_{ij})T_{sj} \end{aligned} \tag{11}$$

where $\mathbf{g}^{(j)}(\Phi_j) = [g^{(j)}(1), g^{(j)}(2), \cdots, g^{(j)}(L)]^{\mathrm{T}}$ and $\Phi_j = \{K_j, T_{1j}, T_{2j}, \tau_j\}$ denote the FIR vector and the parameters of the jth SOPDT model, respectively. The parameters of all of the SOPDT models can be denoted as $\mathbf{\Phi} = [\Phi_1, \cdots, \Phi_m]^{\mathrm{T}}$.

3.2. FIR Length Autoselection. In previous research, the length of FIR (L) is selected manually. This approach has two limitations. First, various L values must be tested in order to find the most appropriate one, which can achieve good enough results and has a reasonable computational cost. Each time a value of L is chosen, a soft sensor model should be trained and evaluated. This will dramatically increase the computational costs.

Second, because the settling time (T_l) values of different process variables are different, secondary variables should have different values of L. In previous studies, identical L is utilized for all secondary variables. If some of the process variables have relatively small settling times, many of their FIR values are close to zero. As a result, the computational cost spent on the calculation of their corresponding intermediate variables are wasted.

In this work, the length of the FIR will be selected automatically. When L is selected manually, the first L values of FIR is supposed to embrace the overall shape of FIR. That is, let L be the cutoff point, and the FIR values from L to ∞ is deemed as values close to zero. A selection standard is defined in the following autoselection procedures.

First, the derivative of IIR in eq 8 is derived as follows:

$$g^{(j)\prime}(t) = \begin{cases} 0 & t < \dfrac{\tau_j}{T_{sj}} \\ \dfrac{K_j}{T_{1j} - T_{2j}}\left[-\dfrac{1}{T_{1j}/T_{sj}}\mathrm{e}^{-\left(t-\tau_j/T_{sj}\right)/(T_{1j}/T_{sj})} \right. & t \geq \dfrac{\tau_j}{T_{sj}} \\ \left. \quad + \dfrac{1}{T_{2j}/T_{sj}}\mathrm{e}^{-\left(t-\tau_j/T_{sj}\right)/\left(T_{2j}/T_{sj}\right)}\right] & \end{cases} \tag{12}$$

Let $g^{(j)\prime}(t) = 0$ when $t \geq \tau_j/T_{sj}$; then, the time corresponding to the maximum value of $g^{(j)}(t)$ can be derived as

$$t_{\max,j} = \frac{T_{1j}T_{2j}}{(T_{1j} - T_{2j})T_{sj}} \ln\left(\frac{T_{1j}}{T_{2j}}\right) + \frac{\tau_j}{T_{sj}} \tag{13}$$

and the maximum value of $g^{(j)}(t)$ can be expressed as

$$g^{(j)}(t_{\max,j}) = \frac{K_j}{T_{1j} - T_{2j}}[\mathrm{e}^{-T_{2j}/(T_{1j}-T_{2j})\ln(T_{1j}/T_{2j})} - \mathrm{e}^{-T_{1j}/(T_{1j}-T_{2j})\ln(T_{1j}/T_{2j})}] \tag{14}$$

If $t > t_{\max,j}$ and $g^{(j)}(t) < \alpha g^{(j)}(t_{\max,j})$, where α is a constant that has a magnitude of $1e - 2$, then $L_j = t$ is considered to be a sufficiently large FIR length for the jth secondary variable.

3.3. Model Training Procedures.

Note that the tuning of the model gain (parameter K) can be covered by the tuning of weights and bias in CNN. Therefore, the value of K in SOPDT models can be taken to be equal to 1 and does not need to be tuned.

In order to train the FIR-CNN model, we have the following procedures:[16]

(1) Secondary variables ought to be selected and preprocessed.

(2) The parameters of FIR, including T_{1j}, T_{2j}, and τ_j, ought to be initialized. By using the impulse response template proposed by Lu et al.,[14] the ranges of those parameters can be determined according to process mechanisms. The parameters in CNN, including the weights and bias in each layer, should also be initialized.

(3) Calculate the values of intermediate variables by process data, and use the outputs of FIRs as the input of CNN. Keep the parameters of FIRs unchanged and train the CNN model.

(4) Calculate the value of the cost function formulated in eq 6. If $J(\mathbf{W}, \mathbf{b}) < \xi$, where ξ is a given accuracy threshold, then the entire model is well-trained. Otherwise, go to the next step.

(5) Keep the parameters of CNN unchanged and optimize the parameters of FIRs by using an intelligent optimization algorithm. In this research, the modified differential evolution (MDE) algorithm proposed by Lu et al.[14] is adopted. The individual with the smallest cost function value is chosen.

(6) Calculate the values of intermediate variables using the FIR parameters optimized in the previous step and train the CNN again.

(7) Repeat steps 4–6 until the entire model is well-trained.

According to the previous derivations, the pseudocode of FIR-CNN can be given as described in Algorithm 1.

Algorithm 1 FIR-CNN

```
Input: U: process data; y: quality data; $\xi$: accuracy threshold;
Output: Trained parameters $\mathbf{\Phi}$, $\mathbf{W}$, $\mathbf{b}$;
Initialization: $\mathbf{\Phi} = \mathbf{\Phi}_0$, $\mathbf{W} = \mathbf{W}_0$, $\mathbf{b} = \mathbf{b}_0$;
Repeat
    Compute values of intermediate variables by eq 11 using the latest $\mathbf{\Phi}$;
    Train CNN to update $\mathbf{W}$ and $\mathbf{b}$;
    Calculate $J(\mathbf{W}, \mathbf{b})$ by eq 6;
    Keep $\mathbf{W}$ and $\mathbf{b}$ unchanged, and update $\mathbf{\Phi}$ by an intelligent algorithm.
Until $(J(\mathbf{W}, \mathbf{b}) < \xi)$
```

4. SIMULATION CASE STUDY

In order to validate the effectiveness of the CNN and FIR-CNN models proposed in this research, a simulation model is studied in this section.

4.1. Process Description and Data Selection.

The simulation model is formulated as

$$\begin{aligned} x_1(s) &= \frac{1}{200s + 1}\mathrm{e}^{-30s}u_1(s) \\ x_2(s) &= \frac{1}{(240s + 1)(90s + 1)}\mathrm{e}^{-60s}u_2(s) \\ x_3(s) &= \frac{1}{(60s + 1)(70s + 1)}\mathrm{e}^{-90s}u_3(s) \\ y(t) &= \ln(x_1(t) + x_2(t) + x_3(t) + 3) + \epsilon(t) \end{aligned} \tag{15}$$

where

$$\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \tag{16}$$

where $w_1(t)$ and $w_2(t)$ are generated from two independent variables with Gaussian distributions that have the same mean and variance: $w_1(t) \sim \mathcal{N}(0, 0.25)$, $w_2(t) \sim \mathcal{N}(0, 0.25)$. The noise $\epsilon(t)$ added is Gaussian white noise: $\epsilon(t) \sim \mathcal{N}(0, \sigma^2)$, where σ is taken to be equal to 0.02, 0.04, and 0.08, respectively. The sampling intervals of the process variables u_j ($j = 1, 2, 3$) are 30 s, while the sampling interval of the quality variable y varies over a range of 20–30 min. The simulation time of the process data is 10 000 s. The training, validation, and test datasets are divided by percentages of 60%, 20%, and 20%, respectively.

4.2. Model Selection and Parameter Settings.

Several models are adopted in this case for comparison, and these models include the following:

(1) Our model: FIR-CNN;

(2) Our model: CNN;

(3) The RNN model, proposed by Su et al.;[30]

(4) The LSTM model, proposed by Ke et al.;[12]

(5) The dynamic ANN (DANN) model, proposed by Du et al.;[31]

(6) The combination of FIR and RNN (FIR-RNN);

(7) The combination of FIR and LSTM (FIR-LSTM); and

(8) The combination of FIR and DANN (FIR-DANN).

The length of the impulse response of the first tunnel is automatically selected in FIR-CNN as 24, 52, and 20, respectively. The time lag (d) in the corresponding CNN model is chosen to be 32, which is the mean value of the autoselected impulse response length in FIR-CNN. Similarly, the time lags in RNN, LSTM, and DANN are chosen as the

DOI: 10.1021/acs.iecr.9b02513
Ind. Eng. Chem. Res. 2019, 58, 11521–11531

Table 1. Parameters of Impulse Response Estimated by FIR-CNN

			Estimated Value		
parameter	input tunnel No.	real value	σ = 0.02	σ = 0.04	σ = 0.08
time constant	1	200	[145.65,61.42]	[166.82,40.02]	[217.45,54.93]
	2	[240, 90]	[303.55,79.96]	[292.03,98.61]	[304.37,101.52]
	3	[60, 70]	[43.46,83.46]	[77.15,87.04]	[50.80,67.78]
delay	1	30	31.85	26.97	28.80
	2	60	44.78	79.97	58.09
	3	90	111.60	95.02	124.28

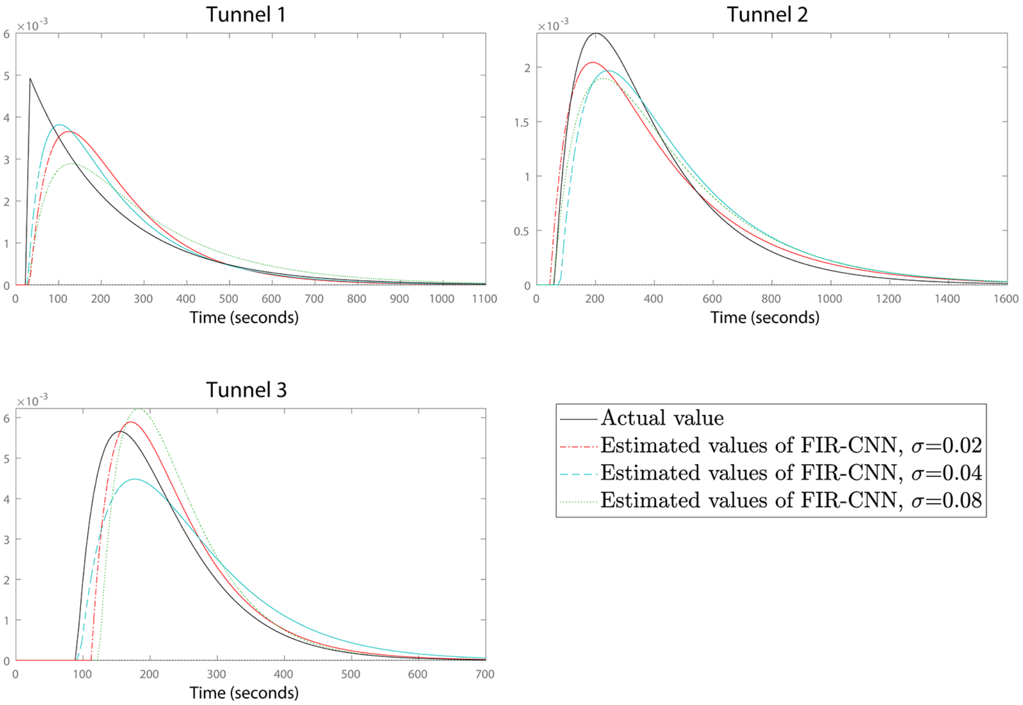


Figure 4. Impulse responses of the SOPDT models.

mean value of the autoselected impulse response length in FIR-RNN, FIR-LSTM, and FIR-DANN, respectively. The number of population in the MDE algorithm is chosen to be 30. The CNN architectures in both proposed soft sensor models are the same: There are two convolutional layers, two pooling layers, and one fully connected layer. The number of convolution kernels is selected to be 2 and 4, respectively. The number of hidden neurons in the fully connected layer is chosen to be 3. In the CNN-based soft sensor model, the heights and strides of kernels in each convolutional layer are chosen to be 2, and the size of pooling windows are taken as 2 in each pooling layer. In FIR-CNN, the heights and strides of kernels are all selected to be 1.

For the sake of fairness, there are two recurrent layers and one fully connected layer in RNN and FIR-RNN. The number of cells and hidden neurons in RNN is set as 2, 4, and 3, respectively. Similarly, there are two LSTM layers plus one fully connected layer in LSTM and FIR-LSTM. The number of units and hidden neurons in LSTM is chosen as 2, 4, and 3, respectively. There are three hidden layers in DANN and FIR-DANN, and the number of hidden neurons in DANN is selected as 2, 4, and 3, respectively. All of those layers mentioned above use the hyperbolic tangent function as the activation function. The RMSProp optimizer is used as the training algorithm used in each model. Early stopping is used in each model, and the training process will be terminated if the validation error begins to increase.

The simulation experiment is performed on a Macintosh computer system (with Intel Core i5 2.9 GHz, 8GB RAM) using Python 3.6. The training times of FIR-CNN and CNN are 20−30 min and 10−20 s, respectively.

4.3. Parameter Estimation. The estimated parameters of those models are listed in Table 1. By utilizing the estimated parameters listed in Table 1, the impulse responses of process variables are plotted in Figure 4.

On one hand, it can be seen from Table 1 that the estimated parameters of tunnel 1 are not close to the real ones. It can also be seen from Figure 4 that the estimated impulse response of tunnel 1 has a relatively large deviation with the real one, compared to the other two tunnels. This is mainly because the first model is a first order plus dead time (FOPDT) model, while we assume that it is a SOPDT model. On the other hand, the estimated parameters of tunnel 2 and 3 are very accurate, regardless of the value that σ takes. In Figure 4, the overall shape of the estimated impulse responses of tunnels 2 and 3 are almost the same as the real ones. This will help FIR-CNN to achieve good performance in predicting the quality variable.

The prediction results corresponding to σ = 0.04 will be adopted for comparison with other soft sensor models.

Prediction Results Analysis. In order to evaluate the performance of different models, the root-mean-squared error (RMSE) values and the correlation coefficient (r) values between the predicted values and real values of the quality data are calculated and listed in Table 2, where $\mathrm{RMSE}_{\mathrm{train}}$ and $\mathrm{RMSE}_{\mathrm{test}}$ represent the RMSE values of training and test datasets, respectively; and r_{train} and r_{test} represent the r values of training and test datasets, respectively. The prediction results of the test dataset by different models are illustrated in Figure 5.

Table 2. RMSE and r Values of Different Models on Simulation Data

	RMSE Values		r Values	
model	$\mathrm{RMSE}_{\mathrm{train}}$	$\mathrm{RMSE}_{\mathrm{test}}$	r_{train}	r_{test}
FIR-CNN	0.0473	0.0498	0.9024	0.9353
CNN	0.0508	0.0525	0.8784	0.8944
FIR-RNN	0.0591	0.0667	0.8386	0.8395
RNN	0.0713	0.0811	0.7394	0.7193
FIR-LSTM	0.0615	0.0670	0.8168	0.8167
LSTM	0.0687	0.0736	0.7589	0.7895
FIR-DANN	0.0693	0.0803	0.7947	0.7741
DANN	0.0457	0.0887	0.9019	0.6434

From Table 2, we can see that the prediction error of FIR-CNN on the test dataset is the minima among all of the models. Besides, the maximum correlation coefficients on both the training and test datasets are achieved by FIR-CNN. From Figure 5, almost all of the data points are precisely predicted by FIR-CNN. This indicates that FIR-CNN not only has strong representation ability but also has strong generalization ability.

CNN outperforms the other models on the test dataset. The predicted values of CNN in Figure 5 are also rather close to the real ones. Many data points in Figure 5 are not accurately predicted by RNN. The correlation coefficients of LSTM are relatively large. However, some data points are not well-predicted. Although the training error of DANN is lower than all of the other models, the test error of DANN is larger than all of the other models. This means overfitting is likely to appear when DANN is applied to this simulation data. In RNN, LSTM, and DANN, the correlations between process variables are not well-utilized, which is one of the reasons that the performance of these three models are worse than FIR-CNN and CNN.

When adding FIRs before these models, the prediction errors of the test data are all reduced, and the correlation coefficients of the test data are all increased. This demonstrates that the information in historical process data can be effectively utilized by the FIRs. However, it can be seen that FIR-CNN still achieves the best performance among the models with FIR, which demonstrates the necessity of utilizing the correlations between process variables by CNN.

5. INDUSTRIAL CASE STUDY: SCGP

In this section, an industrial case study of the Shell Coal Gasification Process (SCGP) is presented to further demonstrate the effectiveness of the proposed models.

5.1. Process Description.

SCGP, whose flow diagram is shown in Figure 6,[32] is a commercial coal gasification system developed by Shell and Krupp-Koppers from the 1970s. There are three primary building blocks in a typical SCGP: the first is the feedstock preparation and feeding block, which includes grinding, drying, and feeding of feedstock (coal, etc.); the second is the gasification block, which mainly includes a gasifier and a syngas cooler; and the third is the gas separation and cleaning block, whose role is to remove dry solids.[33]

The overall production process can be described as follows: pulverized coal, steam, O_2, and N_2 are fed into the gasifier and burned under high-temperature and high-pressure conditions. Then, the liquid slag on the inner wall of the gasifier is exhausted from the bottom of the gasifier. The produced syngas leaves the gasifier and quenched by the quench gas.

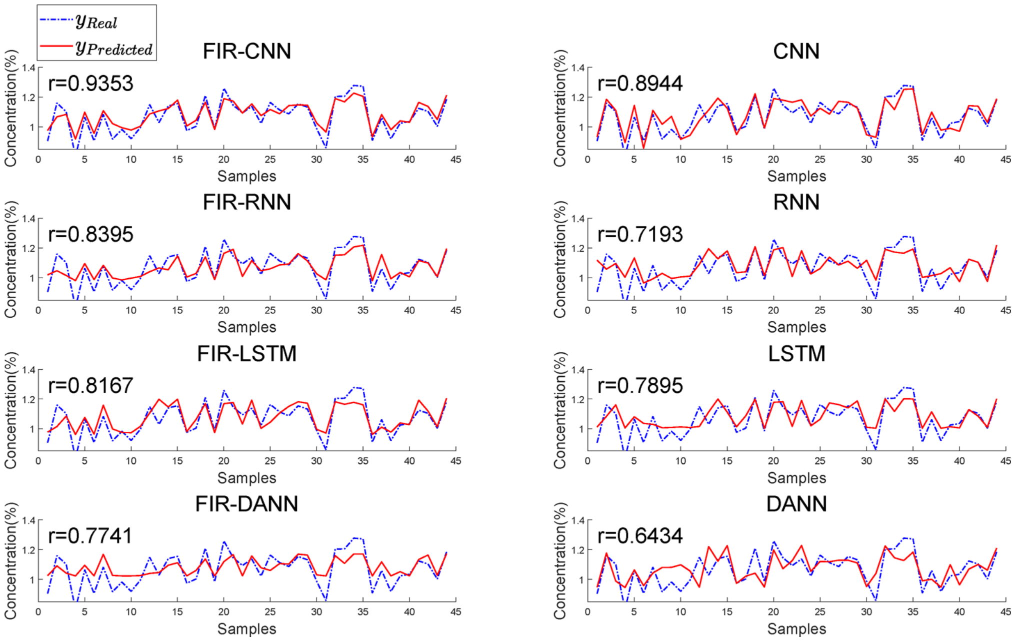


Figure 5. Quality prediction results by different models on simulation data.

DOI: 10.1021/acs.iecr.9b02513
Ind. Eng. Chem. Res. 2019, 58, 11521−11531

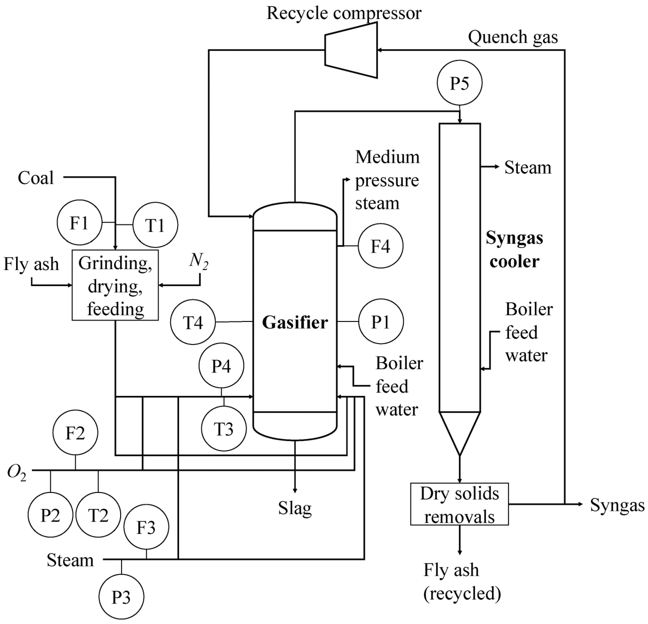


Figure 6. Flow diagram of SCGP.

Later, the syngas is further cooled in the syngas cooler. At last, fly ash is removed and recycled to the feeding block.[34]

The SCGP is a multivariable system with strong nonlinearity and large inertia. According to the mechanism of SCGP, if optimal operating conditions are reached, the concentration of CO_2 will be the lowest. This indicates that the concentration of CO_2 is one of the most essential quality variables that must be controlled. However, online measurement of CO_2 is a difficult task. In order to obtain the off-line value of the concentration of CO_2, fly ash removal and laboratory analysis often take ~40 min. As a result, the analyzed concentration of CO_2 can no longer properly reflect the actual operating conditions, and it is hard to provide accurate information for quality control. Therefore, online estimation of CO_2 concentration by establishing soft sensor models is significant to the stable long-term operation of SCGP.

SCGP is a closed-loop industrial process, and there exist correlations between process variables when SCGP works in a steady-state. Therefore, it is reasonable to utilized CNN to build soft sensor models. Besides, the dynamic characteristics of SCGP can be described by SOPDT models. Therefore, it is also reasonable to add the finite impulse responses of SOPDT models before CNN to extract process dynamics.

5.2. Data Selection.

In this case study, all of the data are collected from a Shell coal gasification unit. The quality variable is selected as CO_2 concentration, whose sampling interval is 1 min. By analyzing the mechanism of SCGP, 14 variables, which are listed in Table 3 and labeled in Figure 6, are selected as secondary variables. The sampling intervals of those variables are 1 min. The training, validation, and test datasets are selected in the time order by the percentages of 60%, 20%, and 20%, respectively.

5.3. Parameter Settings.

The same five models adopted in the previous section will be applied in this process. In FIR-CNN, FIR-RNN, FIR-LSTM, and FIR-DANN, the lengths of the impulse response are automatically selected, while the time lags in the corresponding models are chosen as the mean of autoselected impulse response length in FIR-CNN. The population number in the MDE algorithm is chosen to be 50. There are three convolutional layers, two pooling layers, and two fully connected layers in the CNN-based soft sensor model. The number of convolution kernels is selected as 15, 20, and 25, respectively, and the heights and strides of kernels in each convolutional layer are chosen to be 2. The number of hidden neurons in the fully connected layer is chosen as 15 and 5, respectively. The sizes of pooling windows are taken as 2 in each pooling layer. For the sake of fairness, the CNN architecture in the FIR-CNN-based soft sensor model is almost the same as that observed for CNN, except for two details: (i) the heights and strides of kernels are all selected as 1, and (ii) there is no pooling layer in FIR-CNN.

Table 3. Selected Process Variables of SCGP

No.	tag	variable description	unit
1	F1	flow rate of the coal feed	kg/s
2	F2	flow rate of the O_2 feed	kg/s
3	F3	flow rate of the steam feed	kg/s
4	F4	flow rate of the steam generated from a medium-pressure steam drum	t/h
5	T1	temperature of the coal feed	°C
6	T2	temperature of the O_2 feed	°C
7	T3	temperature of the O_2 and steam mixture in the burners	°C
8	T4	temperature of the membrane wall	°C
9	P1	pressure of the gasifier	MPa
10	P2	pressure of the O_2 feed	MPa
11	P3	pressure of the steam feed	MPa
12	P1−P4	differential pressure of the coal between the burners and the gasifier chamber	MPa
13	P5	pressure at the inlet of the syngas cooler	MPa
14	F2/F1	O_2:coal ratio	

There are three recurrent layers and two fully connected layers in RNN and FIR-RNN. The number of cells and hidden neurons in RNN is set as 15, 20, 25, 15, and 5, respectively. Similarly, there are three LSTM layers plus two fully connected layers in LSTM and FIR-LSTM. The number of units and hidden neurons in LSTM is also chosen as 15, 20, 25, 15, and 5, respectively. There are five hidden layers in DANN and FIR-DANN, and the number of hidden neurons in DANN is selected as 15, 20, 25, 15, and 5, respectively. All of those layers mentioned above use the hyperbolic tangent function as the activation function. The RMSProp optimizer is used as the training algorithm in each model, and early stopping is utilized in each model.

5.4. Prediction Results Analysis.

The RMSE and correlation coefficient (r) values of different models are listed in Table 4. The prediction results of different models are shown in Figure 7.

Table 4 shows that FIR-CNN achieves the best performance on the test dataset. Besides, the trend of predicted values is consistent with the real one. This means FIR-CNN is able to give a reliable trend of CO_2 to operators in real time, and operators can thus change the operating condition of SCGP. Moreover, it can be seen from Figure 7 that the variations of the predicted values of FIR-CNN are consistent with the real data. Actually, there are rarely severe variations in a process dynamic behavior if the process operates under normal conditions.[35] Therefore, the trend predicted by FIR-CNN is relatively reliable.

Note that there are many reasons why the prediction results of FIR-CNN do not seem to be very accurate. First, the

Table 4. RMSE and *r* Values of Different Models on SCGP Data

	RMSE Values		*r* Values	
model	$RMSE_{train}$	$RMSE_{test}$	r_{train}	r_{test}
FIR-CNN	0.0542	0.1021	0.8832	0.6237
CNN	0.0535	0.1117	0.8541	0.5515
FIR-RNN	0.0347	0.1038	0.9464	0.4348
RNN	0.0505	0.1186	0.8801	0.4337
FIR-LSTM	0.0415	0.1165	0.9211	0.5577
LSTM	0.0307	0.1479	0.9638	0.5504
FIR-DANN	0.0319	0.1217	0.9540	0.5781
DANN	0.0475	0.1389	0.8960	0.4269

training, validation, and test datasets are selected in the time order, rather than selected randomly. This is because the soft sensor models are trained off-line and used to make online quality predictions. Since industrial process data are time series data, it is impossible to randomly choose some industrial data in the future to train a soft sensor model. Therefore, it is reasonable to select these datasets in the time order, even if the quality prediction results are much worse than selecting these datasets randomly. Second, the prediction result on the test dataset with $r = 0.6237$ is quite good, because there exist large process noises in the process variables of SCGP. Moreover, the reaction temperature, which is closely related to the CO_2 concentration, is unmeasurable. Consequently, the performance of soft sensor models will inevitably degrade. Actually, the trend predicted by FIR-CNN is much more accurate than the other models. Because of the existence of feedback control, the quality of the products can be stabilized, even though the prediction accuracy is not very high. Therefore, the prediction results of FIR-CNN are good enough for the control of CO_2 concentration.

CNN performs better than the other models without FIR. However, the values predicted by CNN vary much severer than the real values. The main reason is that CNN lacks process mechanisms. If process data contain strong dynamic features, CNN is unable to describe the dynamics of SCGP accurately, and the performance of CNN will, therefore, degrade greatly. The variations of the predicted values of RNN are also obviously higher than the real data. This means RNN is not capable of revealing the dynamic behavior of SCGP. Although LSTM achieves the lowest training error, the test error of LSTM is the largest among all of the models. It can be seen from Figure 7 that the differences between the values predicted by LSTM and real values are relatively large. Thus, overfitting may exist in this LSTM model, which can lead to unsatisfactory prediction results. Although the training error of DANN is lower than that of FIR-CNN and CNN, the test error of DANN is relatively large. From Figure 7, we can see that the prediction values of DANN also have fast variations. This means that it is inappropriate to apply DANN in this process. When adding the FIRs before RNN, LSTM, and DANN, the performances of these models improve on both training and test datasets. This shows the effectiveness of the combination of FIR and normal soft sensor models. Note that FIR-CNN still outperforms the other models, which demonstrates that taking advantage of the correlations between process variables by CNN is essential for quality prediction.

Note that, in this industrial case, a large number of process data are utilized to build soft sensor models in order to extract dynamic features of the process. As a result, the number of parameters of the models will be dramatically large. Table 5 lists the number of parameters to be trained in different models.

In FIR-CNN, abundant process data are utilized by FIR, and the model complexity is greatly reduced by the convolution of FIR and process data. In CNN, the model complexity can also be significantly reduced by increasing the heights and strides of the convolution kernels. Compared to FIR-CNN and CNN, those process data are not appropriately utilized in RNN, LSTM, and DANN. There is no dimensionality reduction measure in these three models. As a result, the structure of LSTM and DANN models are much more complicated than FIR-CNN and CNN. This is one of the reasons why the

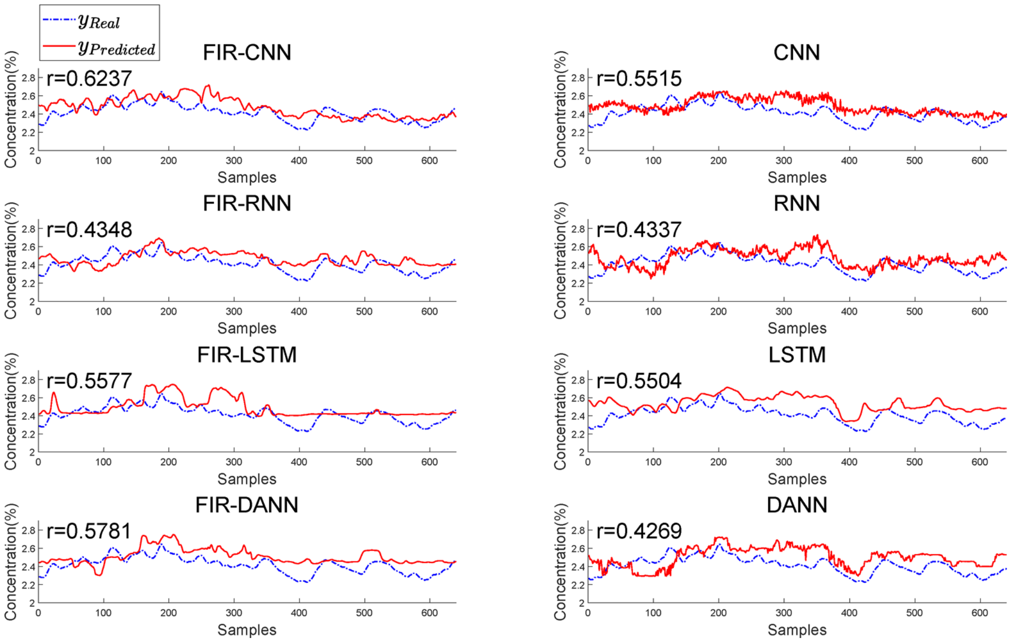


Figure 7. Quality prediction results by different models on the SCGP process.

DOI: 10.1021/acs.iecr.9b02513

Table 5. Number of Parameters To Be Trained

model	number of trainable parameters
FIR-CNN	1546
CNN	2556
FIR-RNN	2796
RNN	2796
FIR-LSTM	9756
LSTM	9756
FIR-DANN	1546
DANN	9106

performances of LSTM and DANN are unsatisfactory. The model complexity of DANN is greatly reduced by adding FIRs. Therefore, FIR-DANN outperforms DANN in previous case studies.

6. CONCLUSION

Two CNN-based soft sensor models are proposed in this model. The first model is a special case of conventional CNN. By performing convolution operations on process data, correlations between process variables can be embraced. By adjusting the parameters of CNN, the model complexity can be greatly simplified. The second model is built by combining FIR with the first model. Historical process data can be reasonably utilized by FIR, and the model complexity is further reduced. The proposed two models outperform the other models in the two case studies. Moreover, FIR-CNN can provide operators with a reliable trend and relatively accurate prediction values of essential quality-relevant variables. Therefore, FIR-CNN has more application potential, compared to other dynamic soft sensor models.

■ AUTHOR INFORMATION

Corresponding Author

*Tel.: 010-62786294. Fax: 010-62770325. E-mail: yangfan@tsinghua.edu.cn.

ORCID

Dexian Huang: 0000-0001-7743-0023

Fan Yang: 0000-0002-0254-9000

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors appreciate the funding from the National Natural Science Foundation of China (Nos. 61673236 and 61873142), the National Basic Research Program of China (No. 2012CB720505), and the Seventh Framework Programme of the European Union (No. P7-PEOPLE-2013-IRSES-612230). The authors also appreciate the help of Dr. Wenxiang Lv and Dr. Xinqing Gao.

■ REFERENCES

(1) Ge, Z.; Huang, B.; Song, Z. Mixture semisupervised principal component regression model and soft sensor application. *AIChE J.* **2014**, *60*, 533−545.

(2) Brosilow, C.; Tong, M. Inferential control of processes: Part II. The structure and dynamics of inferential control systems. *AIChE J.* **1978**, *24*, 492−500.

(3) Yuan, X.; Ge, Z.; Song, Z. Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes. *Ind. Eng. Chem. Res.* **2014**, *53*, 13736−13749.

(4) Zamprogna, E.; Barolo, M.; Seborg, D. E. Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis. *J. Process Control* **2005**, *15*, 39−52.

(5) Sharmin, R.; Sundararaj, U.; Shah, S.; Vande Griend, L.; Sun, Y.-J. Inferential sensors for estimation of polymer quality parameters: Industrial application of a PLS-based soft sensor for a LDPE plant. *Chem. Eng. Sci.* **2006**, *61*, 6372−6384.

(6) Dayal, B. S.; Macgregor, J. F. Recursive exponentially weighted PLS and its applications to adaptive control and prediction. *J. Process Control* **1997**, *7*, 169−179.

(7) Fortuna, L.; Graziani, S.; Xibilia, M. G. Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Engineering Practice* **2005**, *13*, 499−508.

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

(9) Ku, W.; Storer, R. H.; Georgakis, C. Disturbance detection and isolation by dynamic principal component analysis. *Chemom. Intell. Lab. Syst.* **1995**, *30*, 179−196.

(10) Kaspar, M. H.; Ray, W. H. Chemometric methods for process monitoring and high-performance controller design. *AIChE J.* **1992**, *38*, 1593−1608.

(11) Su, H. T.; McAvoy, T. J.; Werbos, P. Long-term predictions of chemical processes using recurrent neural networks: A parallel training approach. *Ind. Eng. Chem. Res.* **1992**, *31*, 1338−1352.

(12) Ke, W.; Huang, D.; Yang, F.; Jiang, Y. Soft sensor development and applications based on LSTM in deep neural network. In *Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, USA, 2017; pp 1−6.

(13) Ma, Y.; Huang, D.; Jin, Y. Discussion about dynamic soft sensing modeling. *J. Chem. Ind. Eng.* **2005**, *56*, 1516−1519.

(14) Lu, W.; Yang, Q.; Huang, D.; Jin, Y. A dynamic soft-sensing method based on impulses response template and parameter estimation with modified DE optimization. In *Proceedings of the 17th International Federation of Automatic Control (IFAC) World Congress*, Seoul, Korea, 2008; pp 10983−10988.

(15) Shang, C.; Gao, X.; Yang, F.; Huang, D. Novel Bayesian framework for dynamic soft sensor based on support vector machine with finite impulse response. *IEEE Trans. Control Syst. Technol.* **2014**, *22*, 1550−1557.

(16) Gao, X.; Yang, F.; Huang, D.; Ding, Y. An iterative two-level optimization method for the modeling of Wiener structure nonlinear dynamic soft sensors. *Ind. Eng. Chem. Res.* **2014**, *53*, 1172−1178.

(17) Hinton, G. E.; Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504−507.

(18) 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.

(19) Yao, L.; Ge, Z. Deep learning of semisupervised process data with hierarchical extreme learning machine and soft sensor application. *IEEE Transactions on Industrial Electronics* **2018**, *65*, 1490−1498.

(20) 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 Transactions on Industrial Informatics* **2018**, *14*, 3235−3243.

(21) Le Cun, Y.; Jackel, L.; Boser, B.; Denker, J.; Graf, H.; Guyon, I.; Henderson, D.; Howard, R.; Hubbard, W. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine* **1989**, *27*, 41−46.

(22) Abdel-Hamid, O.; Mohamed, A.-r.; Jiang, H.; Deng, L.; Penn, G.; Yu, D. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2014**, *22*, 1533−1545.

(23) Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Machine Learning Res.* **2011**, *12*, 2493−2537.

(24) Lee, D.; Siu, V.; Cruz, R.; Yetman, C. Convolutional neural net and bearing fault analysis. In *Proceedings of the International Conference on Data Mining Series (ICDM)*, Barcelona, Spain, 2016; pp 194−200.

(25) Lee, K. B.; Cheon, S.; Kim, C. O. A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing* **2017**, *30*, 135−142.

(26) Zhu, W.; Ma, Y.; Zhou, Y.; Benton, M.; Romagnoli, J. Deep learning based soft sensor and its application on a pyrolysis reactor for compositions predictions of gas phase components. *Comput.-Aided Chem. Eng.* **2018**, *44*, 2245−2250.

(27) LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278−2324.

(28) Wu, H.; Zhao, J. Deep convolutional neural network model based chemical process fault diagnosis. *Comput. Chem. Eng.* **2018**, *115*, 185−197.

(29) Wang, K.; Shang, C.; Yang, F.; Jiang, Y.; Huang, D. Automatic hyper-parameter tuning for soft sensor modeling based on dynamic deep neural network. In *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017; pp 989−994 (DOI: 10.1109/SMC.2017.8122739).

(30) Su, H. B.; Fan, L.; Schlup, J. R. Monitoring the process of curing of epoxy/graphite fiber composites with a recurrent neural network as a soft sensor. *Engineering Applications of Artificial Intelligence* **1998**, *11*, 293−306.

(31) Du, D.; Wu, C.; Luo, X.; Zuo, X. Delay time identification and dynamic characteristics study on ANN soft sensor. In *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, Jinan, China, 2006; pp 42−45.

(32) Bell, D. A.; Towler, B. F.; Fan, M. *Coal Gasification and Its Applications*; William Andrew, 2010.

(33) Mondal, P.; Dang, G.; Garg, M. Syngas production through gasification and cleanup for downstream applications−Recent developments. *Fuel Process. Technol.* **2011**, *92*, 1395−1410.

(34) Monaghan, R. F. D. Dynamic Reduced Order Modeling of Entrained Flow Gasifiers, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2010.

(35) Shang, C.; Huang, X.; Suykens, J. A. K.; Huang, D. Enhancing dynamic soft sensors based on DPLS: A temporal smoothness regularization approach. *J. Process Control* **2015**, *28*, 17−26.