A Novel Soft Sensor Modeling Approach Based on Difference-LSTM for Complex Industrial Process

Jiayi Zhou, Xiaoli Wang, Chunhua Yang, *Senior Member, IEEE*, and Wei Xiong

***Abstract*—The main purpose of soft sensor modeling is to capture the dynamic nonlinear features between the easy-to-measure auxiliary variables and the difficult-to-measure key variables. However, in complex industrial process, it is a challenging work due to the too complicated relationships among the process variables and the base measurement problems. Recently, long short-term memory (LSTM) network shows powerful long-term feature extraction capabilities in complex industrial processes. LSTM focuses on the relationship between the input time series and the output. However, what we concern with are the impact of changes in secondary variables over time on the being detected key variables. In this article, a novel soft sensor modeling approach called difference long short-term memory network is proposed for key variables prediction in complex industrial process. In the method, dynamic information of the inputs is introduced to build a new network unit. Thus, the dynamic temporal features in difference variable and nonlinear features in sequential data are merged to improve the prediction performance. Effectiveness and superiority of the method are validated through detection of the particle size index for a grinding-classification process by comparing to other popular methods.**

***Index Terms*—Complex industrial process, deep learning, difference long short-term memory network (DLSTM), soft sensor.**

I. Introduction

In COMPLEX industrial processes, many key variables are difficult to measure in real time due to expensive detecting instruments or hash measurement environment. Long time of the offline detection leads to delay of the process control and optimization. Therefore, soft sensor has been extensively studied and used in chemical processes, modern power systems, metallurgical processes, and so on [1]–[4].

The main purpose of soft sensor modeling for complex industrial processes is to capture the dynamic nonlinear features between the key variables and the easy-to-measure variables, which are highly correlated with the key variables. It can provide rich information for process monitoring and control. Soft sensor modeling is mainly based on two kinds of models: first-principle models and data-driven models. First-principle models combine prior mechanistic knowledge and mathematical model. However, it is too difficult to obtain accurate mathematical model for complex industrial process because of strong nonlinearity, time-variability, and especially uncertainty (including unknown of the process mechanism, unmeasurement of some process variables, etc.). In comparison to first-principle models, the main superiority of data-driven models is that less mechanical knowledge is involved.

With the extraordinary advances in computing and storage technology, a large number of process data and equipment status data can be obtained from industrial process. Thus, data-driven modeling has become the most popular soft sensor modeling method. The commonly used date-driven modeling methods are partial least squares (PLS)[5], principal component regression (PCR)[6], artificial neural network (ANN) [7], deep learning, etc. In real industrial process, the PLS and the PCR have more applications due to their simplicity[8], [9]. However, since many industrial processes exhibit strong dynamics and nonlinearities, these two methods show a limited ability to express the process features. Theoretically, ANN can approximate arbitrary nonlinear function[10], so back propagation (BP) neural network, extreme learning machines are widely used in the soft sensor modeling research [11], [12]. However, with the increase of network layers, ANN is easy to get stuck in local optima, which restrict its application. In order to track the dynamic characteristics of complex industrial processes, some online update strategies such as just-in-time framework [13], [14] were proposed, but these methods require strong real-time performance, and it is difficult to detect process dynamics and obtain the updated samples. With the introduction of unsupervised layerwise pretraining and supervised fine-tune techniques, deep learning has become a powerful tool recently in many fields, such as computer vision, natural language processing, time series prediction, etc. In recent

Manuscript received November 8, 2020; revised March 21, 2021 and June 4, 2021; accepted August 9, 2021. Date of publication September 8, 2021; date of current version February 2, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62073342, in part by the Basic Science Research Center Program of National Natural Science Foundation of China under Grant 61988101, in part by the Fundamental Research Funds for the Central Universities of Central South University under Grant 2021zzts0181, and in part by the Open Foundation of State Key Laboratory of Process Automation in Mining, and Metallurgy under Grant BGRIMM-KZSKL-2019-09. Paper no. TII-20-5112. *(Corresponding author: Xiaoli Wang.)*

Jiayi Zhou, Xiaoli Wang, and Chunhua Yang are with the School of Automation, Central South University, Changsha 410083, China (e-mail: jiayizhou@csu.edu.cn; xlwang@csu.edu.cn; ychh@csu.edu.cn).

Wei Xiong is with the Changsha Research Institute of Mining and Metallurgy, Changsha 410012, China (e-mail: xiongwei0613@163.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3110507.

Digital Object Identifier 10.1109/TII.2021.3110507

years, a lot of historical data were produced in complex industrial process, which provides more possibilities to build a model that can track the process dynamics and nonlinearities better initially. Therefore, deep learning for soft sensor modeling has attracted many researchers.

In industrial processes, Shang *et al.* [15] introduced a deep belief network for the heavy diesel 95% cut point estimation. This attempt shows the excellent nonlinear features extraction of deep learning in soft sensor modeling. Yan *et al.* [16] built a denoising stacked autoencoders-neural network (DSAE-NN) to estimate the oxygen content. In this model, unsupervised training for DSAE is conducted with unlabeled data. Then, the learned feature is used to fine tune the neural network. But it takes long time to training the network. Yao *et al.* [17] designed a semisupervised hierarchical extreme learning machine to improve the utilization of unlabeled process data. To obtain more information between secondary variables and key variables, Yuan *et al.* [18]–[20] proposed a series of variable-wise weighted stacked autoencoder models. The models take the linear Pearson and nonlinear Spearman correlations of secondary variables and key variables as prior knowledge to initialize the network to learn more quality-relevant features. To capture the uncertainty of industrial process, Khodayar *et al.* [21], [22] proposed a deep learning framework, which incorporates rough set theory to handle the highly varying data source. Zheng *et al.* [23] tried to build a regression generative adversarial network (RGAN) for the prediction of physical properties of crude oil. The discriminator shares its shallow features with the regression model, which is learned from the adversarial learning between discriminator and generator. Then, the loss function of the regression model forces the generator to produce more realistic samples. Results show that RGAN can learn more representation from the data space. Most of these models can handle unlabeled secondary variables result from the long sampling interval of key variables in real industrial processes, but they cannot obtain the temporal features which is more relevant to the corresponding key variables.

In recent years, recurrent neural network (RNN) and its improvements such as long-short term memory (LSTM), gated recurrent unit (GRU) have shown great power in dealing with dynamic industrial time sequence. However, with the length of time sequence increasing, RNN could encounter gradient vanishing or explosion [24]. To overcome these problems, LSTM and GRU were proposed to tackle the long-term dependence of time series. Chou and Wu [25] proposed a sequence-to-sequence model in the form of a nonlinear GRU encoder and GRU decoder in a distillation process from a local refinery. All secondary historical data with or without corresponding key variables are used to learn dynamic feature for the encoder. To embed more physics into hidden state, feature learned from encoder are shared with the decoder. Sun *et al.* [26] presented a probabilistic sequential network, which combines Gaussian–Bernoulli restricted Boltzmann machine (GRBM) and LSTM. The GRBM network is trained in an unsupervised manner. Then, the learned feature is fed into LSTM to achieve dynamic representation. Yuan *et al.* [27] proposed a supervised long short-term memory network (SLSTM), which takes the key variable being detected as input of the network. However, due to the long-term sampling of the key variables, the current key variables may be unknown, which leads to its limited application. Wang *et al.* [28] used pinball loss instead of the mean square error (MSE) to train LSTM for probabilistic load forecasting. Thus, traditional LSTM-based point forecasting is extended to probabilistic forecasting in the form of quantiles. Wen *et al.* [29] conducted a frequency estimation framework based on LSTM for power system. Yuan *et al.* [30], [31] proposed a series attention-based LSTM networks, which combine the spatial attention and temporal attention to identify important input variables that are more related to the key variable at each time and hidden states across all time steps. Compared with traditional data-driven soft sensor models, the excellent dynamic learning ability of LSTM is verified again.

Although LSTM can transfer historical information through memory cell state and hidden neuron cell state of each unit, it cannot reveal the relationship between the dynamic change of the input and the output. For complex industrial processes, most of the time it should be in a stable working state. However, the process is difficult to operate stably for a long time due to the continuous change of raw materials, the state of production equipment and the unknown disturbance of the system, which leads to the continuous change of process parameters. Therefore, it is vital to capture the impact of changes in the inputs on the key variables being detected. Thus, in this article, the difference information of the inputs is introduced to build a new difference long short-term memory network (DLSTM) unit. Both input and difference variables are fed into each DLSTM unit to learn the dynamic latent mapping from the secondary variables to the key variables. Then, DLSTM units are further connected to build deep DLSTM network for soft sensor modeling. To evaluate the proposed model, the experiments with the recent LSTM-based techniques are conducted on the real industrial dataset for grinding classification. The main contributions of this article are summarized as follows:

1) A novel LSTM network with a difference module for soft sensor modeling is developed. The difference structure is designed to introduce the difference information of the inputs to capture the impact of changes in the inputs on the key variables being detected. The difference variables are only fed into the output gate of all layers to learn the dynamic latent mapping between the difference information and the key variables. This structure also can prevent DLSTM from being disturbed by irrelevant information when it learns dynamic features.
2) LSTM model with the difference variables and recent advanced soft sensor models based on deep learning are compared to DLSTM to demonstrate the effectiveness of the module in grinding classification process.

The rest of this article are organized as follows. Section II describes difference-LSTM in detail and the soft sensor modeling based on difference-LSTM. Then, effectiveness of the proposed method is validated through case study in Section III. Finally, Section IV concludes this article.

II. Soft Sensor Modeling Based on DLSTM

In this section, the structure of DLSTM unit and multilayer DLSTM are presented. Then, the procedure for soft sensor modeling based on DLSTM is described.

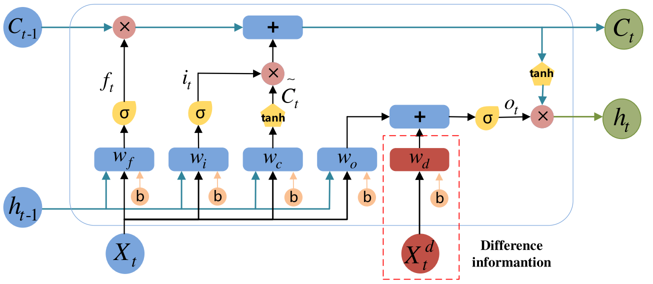


Fig. 1. Structure of the DLSTM unit.

A. Structure of the DLSTM Unit

The basic unit of DLSTM is the LSTM unit block. Although LSTM can transfer dynamic features through memory cell state and hidden neuron cell state of each unit, it cannot reveal the relationship between the dynamic change of the input and the output. For complex industrial processes, most of the time it should be in a stable working state. However, it is difficult for the process to run stably for a long time due to the continuous change of raw materials, the state of production equipment, and the unknown disturbance of the system, which leads to the continuous change of process parameters. Therefore, it is vital to capture the impact of changes in the inputs on the key variables being detected. Hence, DLSTM is proposed to characterize the regularity between input data and key variable by introducing the difference information of the inputs as input feature. The structure of the basic DLSTM unit is shown in Fig. 1. The big difference between LSTM and DLSTM is the inputs of the output gate. For each DLSTM unit, the output gate has three inputs: previous hidden cell state, the current input, and the difference between the previous input and the current input. Thus, the difference features of the inputs can be used to learn actual dynamics of the key variables.

From Fig. 1, the three gates of DLSTM unit are calculated as

$$Inputgate: i_{(t)} = \sigma(W_i \cdot [x_{(t)}, h_{(t-1)}] + b_i) \tag{1}$$

$$Forgetgate: f_{(t)} = \sigma(W_f \cdot [x_{(t)}, h_{(t-1)}] + b_f) \tag{2}$$

$$Outputgate: o_{(t)} = \sigma(W_o \cdot [x_{(t)}, h_{(t-1)}] + W_d \cdot x^d_{(t)} + b_o) \tag{3}$$

$$x^d_{(t)} = x_{(t)} - x_{(t-1)} \tag{4}$$

$\sigma(\cdot)$ notes the activation function. $h_{(t-1)}$ is the previous hidden state. W_i, W_f, W_o are the connection weights of these three gates. b_i, b_f, and b_o are the corresponding bias of the three gates. For better illustration, $x_{(t)}$ in DLSTM is noted as the original input. $x^d_{(t)}$ represents the difference variables, also means the difference information of the original input of DLSTM unit; W_d is the connection weights of the difference variables. Then, the current memory cell and hidden state of DLSTM are updated as

$$\tilde{C_{(t)}} = \sigma_c(W_c \cdot [x_{(t)}, h_{(t-1)}] + b_c) \tag{5}$$

$$C_{(t)} = f_{(t)} \odot C_{(t-1)} + i_{(t)} \odot \tilde{C}_{(t)} \tag{6}$$

$$h_{(t)} = o_{(t)} \odot \sigma_c(C_{(t)}) \tag{7}$$

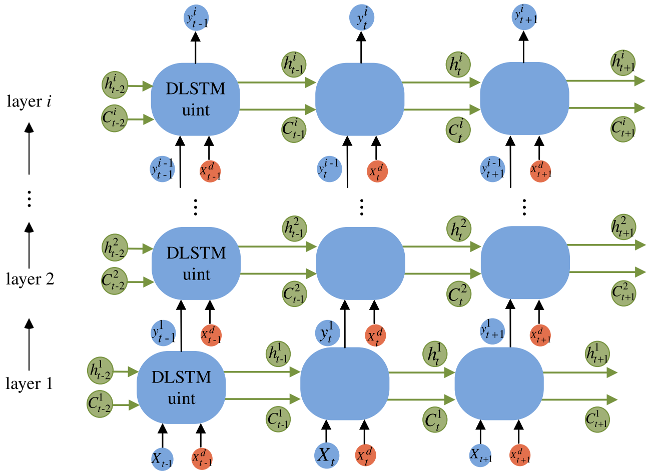


Fig. 2. Structure of the multilayer DLSTM.

where $C_{(t-1)}$ is the previous memory cell state, $\tilde{C}_{(t)}$ represents the input activation vector, $\sigma_c(\cdot)$ is the nonlinear tanh activation function, and $\odot$ denotes the elementwise product operation. Thus, the dynamic difference information of the original inputs can be preserved in the hidden cell state of current time and the memory cell state of next time. DLSTM can handle sequence data by storing the important difference feature in the short-term state and long-term state with this structure.

B. Structure of the Multilayer DLSTM

The basic DLSTM unit can be used to construct a multilayer DLSTM network, the structure of which is provided in Fig. 2. For each unit of layer 1, the total inputs are composed of the current original input and the current difference variables. For latter layers, the total inputs are composed of the output of the previous layer and the difference variables of the previous layer. With this structure, deep nonlinear dynamic difference features can be learned from each layer. Then, the DLSTM network can be trained by the BP through time (BPTT)[32] iteratively.

In order to understand the flow of information in multilayer DLSTM better, the information flow of a three-layer DLSTM network is shown in Fig. 3 to illustrate. The neurons number of the three layers are 150, 10, 10. It is assumed that the dimension of the original input variables and the difference variables are (1, 15). Thus, the total input of DLSTM layer1 are (1, 30). After difference information flow through layer1, the difference representation is obtained by the output with dimension (1,150). Then, it is concatenated with the difference of the original inputs as the total input with dimension (1,165) of the second layer. In the DLSTM unit, the total input will be divided into two parts, which are respectively passed into the corresponding gate. The same goes for the third layer. Then, the final output of DLSTM is fed into a fully connected layer with 1 neuron to predict the result.

C. Modeling Procedure for the Multilayer DLSTM

The main purpose of soft sensor modeling for complex industrial processes is to capture the dynamic nonlinear features

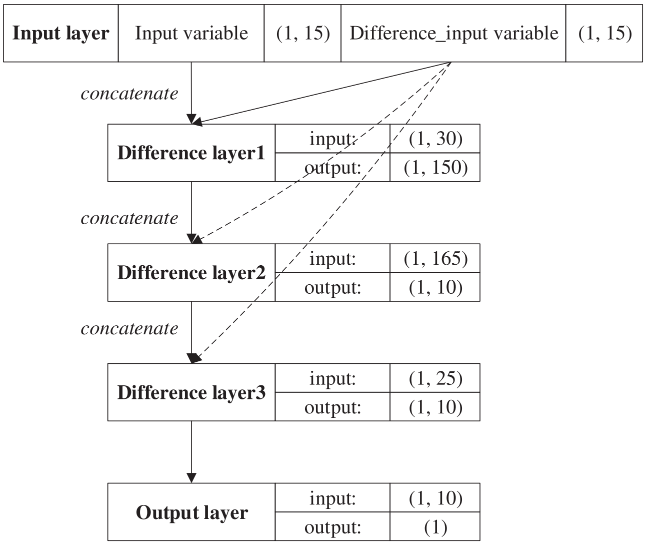


Fig. 3. Information flow of a three-layer DLSTM.

TABLE I
PROCEDURE OF SOFT SENSOR MODELING BASED ON DLSTM

```
Algorithm1: Soft sensor modeling based on DLSTM
Data preparation: Choose secondary variables and time step $Ts$ according to prior knowledge and mechanism of industrial process;
Data preprocess: Convert industrial production data to supervised sequence data, and normalize the data;
Input: A set of supervised data $\{x_{m-(Ts-1)}, x_{m-Ts}, ..., x_m\}$, $\{y_1, y_2, ..., y_n\}$;
Output: Prediction of key variable $\{\hat{y}_1, \hat{y}_2, ..., \hat{y}_n\}$;
Start:
Step1: Determine the training dataset , the validation dataset
       and the testing dataset;
Step2: Determine the layer number and neuron number of
       DLSTM by using trial and error method from a
       candidate set; Choose the optimal structure of DLSTM
       which has the best validation RMSE;
Step3: Initialize all weights $\{W_i, W_f, W_o, W_d\}$ of DLSTM;
Step4: Update $\{W_i, W_f, W_o, W_d\}$ by BPTT algorithm and
       ADAM Optimizer;
Step5: Repeat Step4 under maximum epochs until convergence
       and obtain the early stop model;
Step6: Get the final prediction model for soft sensor application
       to predict the key variable $\{\hat{y}_1, \hat{y}_2, ..., \hat{y}_n\}$ with
       testing samples.
End.
```

between key variables and easy-to-measure variables, which are highly correlated with key variables. The modeling algorithm for DLSTM network is shown in Table I.

Preparation work: The most important task is choosing secondary variables and time step of training and test dataset according to prior knowledge and mechanism of industrial process. Time step of data is generally determined by the response time of a specific industrial process.

Then, the original dataset needs to be serialized as a supervised sequence data. There is a labeled dataset D $= (x_1, x_2, \ldots, x_t), (y_1, y_2, \ldots, y_t) = (x_i, y_i), i = 1, 2, \ldots, t$, where x_i presents the secondary variables and y_i is the key variable, and t is the total number of all data, also means the length of time sequence. After choosing the time step ts of sequence, the serialized dataset $Ds = (x_{m-(ts-1)}, x_{m-ts}, \ldots, x_m)$, (y_n), $m, n \in \{1, 2, \ldots, t\}$. That means using ts past observations of sensor data to predict the key variable y at time n. However, m and n are determined by the mechanism of industrial process and they may not be equal. To reduce data redundancy and improve data integrity, data normalization need to be conducted before sending to DLSTM network. It is normalized to the range of [0 1] for each of the secondary variable by the following expression:

$$x_{normalization} = \frac{(range_{\max} - range_{\min}) \times (x - x_{\min})}{(x_{\max} - x_{\min})} + range_{\min} \tag{8}$$

where $range_{\min} = 0$, $range_{\max} = 1$.

Step 1: Choose an appropriate ratio to divide the dataset into training set, test set, and validation set.

Step 2: Determine the number of neurons and layers of the DLSTM with trial and error technique or prior knowledge and mechanism of industrial process. According to industrial application problems or some specific rules[33] to choose a set of neuron numbers. Then, the network with different neurons will be trained on training set and validated on the validation set. Finally, the optimal trained model will be found based on the best validation root-mean-square error (RMSE). The number of neural network layers is largely related to the size of the dataset. The larger the dataset, the more layers are needed. The procedure to determine the number of layers are the same as the steps to determine the number of neurons.

Step 3: Randomly initialize all weights $\{W_i, W_f, W_o, W_d\}$ of DLSTM.

Step 4: The training loss function is shown as follows:

$$\begin{aligned} \text{Loss} &= \arg\min \text{MSE}(y_{real}, y_{predicted}) \\ &= \arg\min \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \left(y_{real}^i - y_{predicted}^i\right)^2. \end{aligned} \tag{9}$$

Train the DLSTM model by using BPTT algorithm with ADAM optimizer under maximum epochs and obtain the early stop model, Then, the prediction error propagates back along time and network depth. To minimize the loss function by updating the weight coefficients and bias vectors. To avoid overfitting problems, the early stopping is conducted to obtain a model with better generalization performance.

Step 5: Once the DLSTM network is well trained, predict the key variable $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$ with testing samples.

Step 6: At last, the final soft sensor model based on DLSTM is built to predict the key variables.

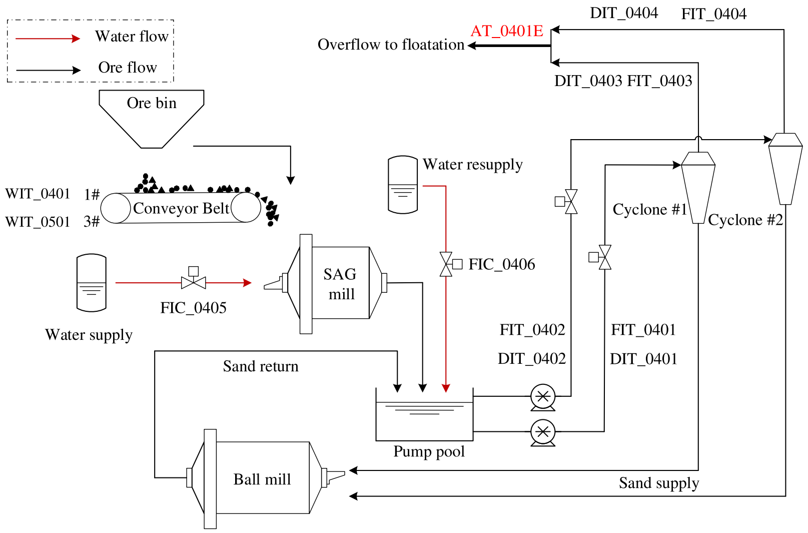


Fig. 4. Flowchart for the grinding classification process.

III. Industrial Case Study: Grinding Classification Process

In this section, the proposed soft sensor model is evaluated by a grinding-classification process to estimate the particle size index of grinding overflow. Then, the effectiveness of the model is compared with other soft sensor models based on deep learning.

A. Process Description

The main purpose of grinding classification process is to produce a pulp with a desired particle size for flotation [34], [35]. Fig. 4 shows the details of a two-stage mill grinding circuit. First, coarse fresh ore on the conveyor belt and a certain amount of water are fed into the semiautogenous (SAG) mill simultaneously to grind. Then, the ore pulp is discharged from the mill into the pump pool. Then with additional water resupply, the pulp is fed into two cyclones for classification. In the cyclones, the pulp including both coarser and finer particles is separated into two streams namely overflow and underflow pulp. The underflow pulp with coarser particles is sent to the ball mill for regrinding. With knocking the steel balls and tumbling in the mill crushes, the underflow pulp get a finer size. Meanwhile, the overflow pulp with finer particles is transported to the floatation. So the particle size of the final overflow pulp determines the quality of flotation production. In other words, the performance of the grinding-classification process plays a determinative role on the technical indices of the whole mineral processing plant. However, the particle size index of the overflow pulp is generally obtained by offline laboratory analysis, which takes a long time and cannot meet the requirements of online optimization control. Hence, a soft sensor modeling method is applied to obtain the particle size index of the overflow pulp.

B. Dataset Preparation

According to prior knowledge and mechanism of grinding classification process, 14 process variables that are highly related with the particle size index of the overflow pulp are selected to be secondary variables. Thus, the particle size index of the overflow pulp is the key quality variable. All variables of soft sensor model are presented in Table II. Specially, historical data of the particle size index also reflect the regularity of change in the particle size. So, the previous particle size of the overflow pulp is taken as an additional auxiliary variable.

TABLE II
Description of the 15 Process Variables in Grinding Classification Process

NO.	TAGS	DESCRIPTION	UNIT
1	WIT_0201	Ore weight of coarse crushing	t/h
2	WIT_0401	Ore weight of Belt 1#	t/h
3	WIT_0501	Ore weight of Belt 3#	t/h
4	DIT_0401	Feed pulp concentration of cyclone #1	%
5	DIT_0402	Feed pulp concentration of cyclone #2	%
6	DIT_0403	Overflow pulp concentration of cyclone #1	%
7	DIT_0404	Overflow pulp concentration of cyclone #2	%
8	FIT_0401	Feed pulp flow of cyclone #1	m^3/h
9	FIT_0402	Feed pulp flow of cyclone #2	m^3/h
10	FIT_0403	Overflow pulp flow of cyclone #1	m^3/h
11	FIT_0404	Overflow pulp flow of cyclone #2	m^3/h
12	FIC_0405	Feed water flow of SAG mill	m^3/h
13	FIC_0406	Add water flow of pump pool	m^3/h
14	FIC_0410	Add water flow for underflow pulp of cyclone	m^3/h
15	AT_0401E	Particle size index of overflow pulp	%

The time step of the model is determined to be 1, since the total response time of the grinding mills and cyclones is almost an hour. An eleven-month period hourly data from December 17, 2015 to November 28, 2016 was collected to sample into all variables. 8352 samples were available. So the total dataset can be expressed as follows:

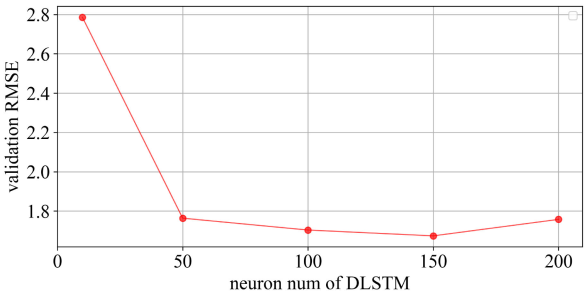


Fig. 5. Validation RMSE of DLSTM for different neuron numbers.

TABLE III
STRUCTURE AND PARAMETERS OF ALL MODELS

Model	Layers number	Neurons number	Input data
DLSTM	3	150	$\{(x_{n-1}, y_{n-1}), (x_{n-1}^d, y_{n-1}^d)\}$
LSTM	3	10	$\{(x_{n-1}, y_{n-1})\}$
LSTM(d)	3	10	$\{(x_{n-1}, y_{n-1}, x_{n-1}^d, y_{n-1}^d)\}$
RNN	3	10	$\{(x_{n-1}, y_{n-1})\}$
GRU	3	10	$\{(x_{n-1}, y_{n-1})\}$
VA-LSTM	3	50	$\{(x_{n-1}, y_{n-1})\}$
RAE	3	20	$\{(x_{n-1}, y_{n-1})\}$
IPDL	3	20	$\{(x_{n-1}, y_{n-1})\}$

Original dataset: $D_{grinding} = \{(x_i, y_i)\}$, $i \in \{1, 2, \ldots, t\}$, $t = 8352$.

Secondary variables: $x_i = [x_i^1, x_i^2, \ldots, x_i^s]$, $s = 14$.

Difference variables: $x_i^d = [x_i^{d1}, x_i^{d2}, \ldots, x_i^{ds}], i \in \{2, \ldots, t\}, x_i^{ds} = x_i^s - x_{i-1}^s$.

Key variable: $y_i, i \in \{1, 2, \ldots, t\}, t = 8352$.

Serialized dataset: $D_{s\_grinding} = \{(x_{n-1}, y_{n-1}), (x_{n-1}^d, y_{n-1}^d), (y_n)\}$.

$n \in \{3, \ldots, t\}, t = 8352$.

Therefore, there are 8350 samples available. All samples is divided into three parts: 70% for training dataset (5880 samples, from December 17, 2015 to August 17, 2016) and 30% for testing dataset (2470 samples, from August 17, 2016 to November 28, 2016). 30% of the training set is the validation set (1764 samples, from June 5, 2016 to August 17, 2016).

C. DLSTM Architecture and Comparison Models

To build an optimal DLSTM model, the number of hidden layers and hidden neurons should be determined. By training the DLSTM models with two hidden layers and three hidden layers separately, it is found that these models are prone to overfitting. Due to the amount of dataset, DLSTM models with more than one hidden layer will have more complex architecture which are easy to get overfitting issue. Therefore, the DLSTM model are composed of a three-layer network, an input layer, a DLSTM layer, and an output layer. In order to select the optimal number of hidden units, DLSTM model is trained and validated by trial and error method from the candidate set 10, 50, 100, 150, 200. The validation RMSE for different neuron numbers are shown in Fig. 5. When the neuron number equals 150, DLSTM model achieves the best RMSE of validation set. Thus, the optimal structure of DLSTM has one hidden layer with 150 hidden units. The learning rate and the batch size are set as 0.001 and 36. The maximum iteration epoch is set 500. To avoid overfitting, the training stops when the validation loss no longer decreases in ten consecutive training epochs.

To evaluate the effectiveness of DLSTM, comparison with seven models is conducted in this case. They are shown as follows:

1) The soft sensor model based on the traditional LSTM.
2) The soft sensor model based on the traditional LSTM, which combines the difference variables and the second variables as one input, it is noted as LSTM(d). The difference variables in LSTM are combined with secondary variables and sent into the network. The difference variables in DLSTM are fed separately into the network as difference information to measure the influence of difference information on the change of the key variable.
3) The soft sensor model based on RNN.
4) The soft sensor model based on GRU.
5) The soft sensor model based on variable attention-based long short-term memory network (VA-LSTM) [29], VA-LSTM can obtain the quality-related inputs through an attention mechanism.
6) The soft sensor model based on rough stacked autoencoder (RAE)[21], RAE can obtain the uncertainty of industrial process by combining deep learning and rough set theory.
7) The soft sensor model based on interval probability distribution learning (IPDL)[22], IPDL can capture interval knowledge from the data based on generative deep learning;

The detailed structure and parameters of all models are shown in Table III. Due to the amount of dataset, all other models are also composed of a three-layer network, an input layer, a hidden layer or one RAE/IPDL unit, and an output layer. The neuron number of LSTM/LSTM(d)/RNN/GRU are selected from the candidate set which is the same as of DLSTM. The neuron number of RAE and IPDL are selected from the set {10, 20, 30, 40, 50}. The maximum iteration epoch is considered to set 500 to train all models, including the unsupervised learning phase of RAE and IPDL. Stochastic gradient descent is adopted to finetune RAE and IPDL models. The learning rate η is set to 1 and the coefficient of the momentum term σ is set to 0.5. The L2 regularization term is conducted to add a complexity penalty and prevent overfitting. The weight decay parameter for the L2 regularization λ is set to 0.001. The simulation processes are accomplished in a personal computer in Python3.6, 64 bit operating system, 4.00 GB of RAM, and Intel(R) Core (7 M) i7-5500 CPU@2.40GHZ 2.40GHZ. To compare the effects of all models in this article, the data of other models are also standardized by min–max normalization.

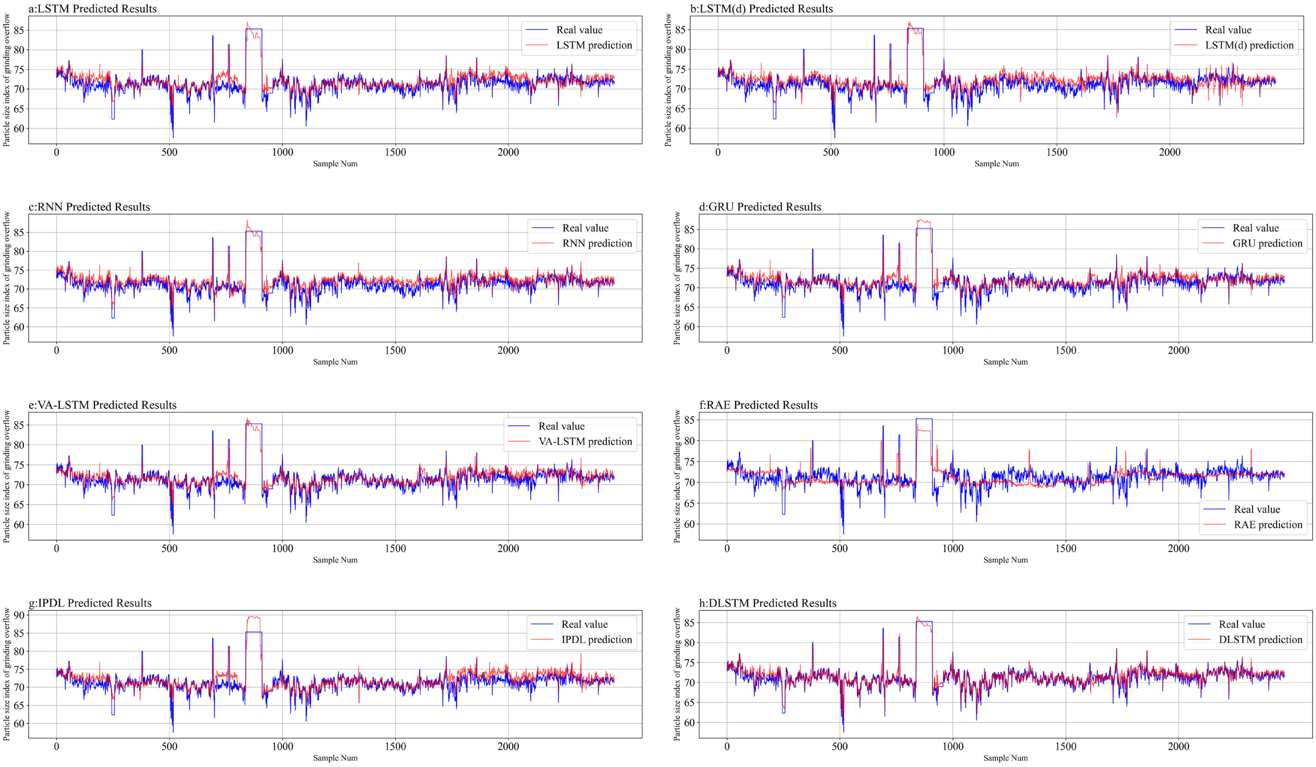


Fig. 6. Predicted results of all models. (a) LSTM. (b) LSTM(d). (c) RNN. (d) GRU. (e) VA-LSTM. (f) RAE. (g) IPDL. (h) DLSTM.

D. Evaluation

The RMSE, the coefficient of determination R^2, and the mean absolute error MAE are used to evaluate the performance of the proposed model. They are defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_{test}} \sum\nolimits_{i=1}^{N_{test}} (y_i - \hat{y}_i)^2} \tag{10}$$

$$R^2 = 1 - \sum_{\mathrm{i}=1}^{N_{test}} (y_i - \hat{y}_i)^2 / \sum_{\mathrm{i}=1}^{N_{test}} (y_i - \bar{y}_i)^2 \tag{11}$$

$$\mathrm{MAE} = \frac{1}{N_{test}} \sum_{\mathrm{i}=1}^{N_{test}} |y_i - \hat{y}_i| \tag{12}$$

where N_{test} is the number of test samples, y_i indicates the real value of the ith testing sample $\hat{y}_i$, $\bar{y}_i$, respectively, represents the predicted value and the average of all predicted values.

TABLE IV
EVALUATION INDICES OF ALL MODELS

Model	RMSE	R^2	MAE
LSTM	1.917	0.638	1.383
LSTM(d)	2.012	0.601	1.525
RNN	1.963	0.620	1.466
GRU	1.876	0.653	1.337
VA-LSTM	1.784	0.686	1.250
RAE	2.359	0.451	1.633
IPDL	2.058	0.582	1.482
DLSTM	**1.622**	**0.741**	**1.090**

E. Results and Discussions

The predicted and real values of all methods for the testing data are displayed in Fig. 6 and the numerical evaluation indices of them are calculated in Table IV. It can be seen from Fig. 6 that the predicted trend of DLSTM is consistent with the real value and the predicted line of DLSTM is much smoother than other models. This shows that the dynamic difference features captured by the DLSTM model are really useful for predicting the output values in a relatively continuous process. This is also reflected in Table IV. Table IV shows that DLSTM achieves the smallest RMSE/MSE on the test dataset. First, the predicted result of LSTM and LSTM(d) shows that the LSTM model performs better than the LSTM(d) model. That does not mean the difference information does not work. The dimension of input data of LSTM(d) is twice that of LSTM, which results in more irrelevant information disturbing the performance of the model. Second, the effectiveness of the difference module can be demonstrated by comparing the LSTM(d) and DLSTM model. The difference variables in LSTM(d) are combined with secondary variables and sent into the network. The difference variables in DLSTM are fed separately into the network as difference information to measure the influence of difference information on the change of the key variable. The main reason why the performance of LSTM(d) is worse than DLSTM is that the difference variables and the secondary variables are fed into the LSTM(d) network without distinction, which leads to the network learn more redundant information. However, with the difference structure, DLSTM can capture more difference features between difference variables and key variables. It also demonstrates the effectiveness of the difference structure in DLSTM. Third, the VA-LSTM model, RAE model, and IPDL model are adopted to further prove the effectiveness of DLSTM. VA-LSTM can obtain the quality-related inputs through an attention mechanism. It can be seen from Table IV, the predicted RMSE of VA-LSTM is 0.162 higher than that of DLSTM, but lower than the other models. It means the variable attention is useful to weight the input variables, which are more relevant to the key variables. However, compared with the VA-LSTM model, the difference variables of DLSTM are more effective to capture the dynamic features from the input variables to the key variables without weight assignment of input variables based on the attention mechanism. RAE and IPDL model achieve unsatisfactory prediction results, the main reason is that there are more other secondary variables interfere with the learning of the models. It can be seen from Table IV, the predicted RMSE of RAE and IPDL is both higher than that of DLSTM. Although they perform excellent prediction for univariate time series, generalization ability decreases when modeling for multivariate time series.

At last, by comparing these five results (LSTM, RNN, GRU, VA-LSTM, DLSTM), it can be found that LSTM/GRU/VA-LSTM/DLSTM has a lower RMSE and a higher R^2 index than RNN. The results reflect the long-term dynamic models perform better than short-term models.

Particularly, it can be seen from Fig. 6 that there is a period of constant value around 800–900 samples. The particle size index of the overflow pulp at this stage has been larger than the previous time. The particle size index of the overflow pulp at this stage has been larger than that before. There are many reasons for this situation. According to the analysis of the industrial process data, insufficient pressure of the cyclone leads to the deterioration of the grinding classification performance of the cyclone. It is very necessary to predict the particle size index in this period. First, it is very important for real-time optimal control of industrial process. The operator can adjust the pressure of the cyclone in time to reach the optimal performance when the soft sensor model predicts this situation. This is exactly what it means to build a soft sensor model for complex industrial process. Second, from the economic point of view, it is essential to control the particle size index of the overflow pulp within the normal range to avoid large fluctuations. If the soft sensor model cannot predict this abnormal situation, the higher particle size index of the overflow pulp will influence the quality of flotation production, which makes more economic loss for the industry.

Fig. 7 shows the scatterplots of the predicted and real values for all methods. The horizontal axis represents the real value of the particle size index of overflow pulp, and the ordinate represents the predicted value of each model. From Fig. 7, it is pretty obvious that the prediction points of DLSTM lie much tighter between [65,75] than other models. However, the performance of all methods degrades significantly when dealing with large or small values. The main reason is that there are relatively few data with larger or smaller key variables in the training set. Thus, all network models cannot learn the corresponding mapping relationship sufficiently. However, in the real industrial process, it will happen when the system works not well. From a security and economic perspective, this is not allowed for a long time. Therefore, it is vital to collect a lot of historical data in complex industrial process to learn dynamic features under different working conditions.

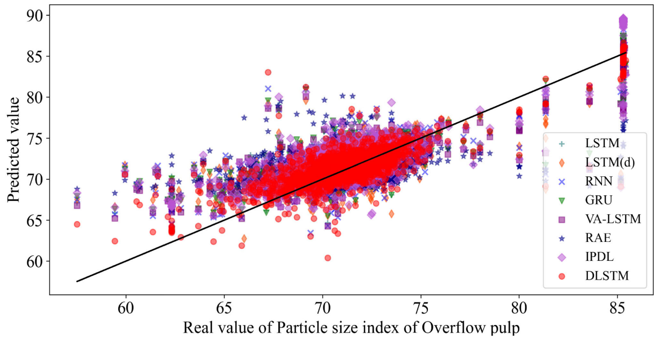


Fig. 7. Predicted scatter plots of all models.

TABLE V
TRAINING TIME AND NUMBER OF PARAMETERS FOR ALL MODELS

Model	Trianing time(s)	Num. of parameters
LSTM	237	1051
LSTM(d)	242	1651
RNN	**136**	271
GRU	210	791
VA-LSTM	218	17216
RAE	—	1441
IPDL	—	716
DLSTM	**156**	**104100**

The distributions of the prediction errors for all models are shown in Fig. 8. The red curve is the fitting Gaussian distribution curve of the prediction error. It can be seen from Fig. 8 that the average value of the prediction error of DLSTM is the smallest, which also indicates that the DLSTM has a more stable prediction ability. However, some individual prediction error of the DLSTM model is more than 5, but the prediction result on the test dataset with $R^2 = 0.741$ is quite good, because there exist large process noises in the process variables of the grinding classification process.

The training time and the number of parameters for all five models are shown in Table V. It can be seen that number of parameters for all models are 1051, 1651, 271, 791, 17 216, 1441, 716, and 104 100. However, due to the unsupervised learning, the training of RAE and IPDL is no longer under the same conditions with other models. Thus, their training time is

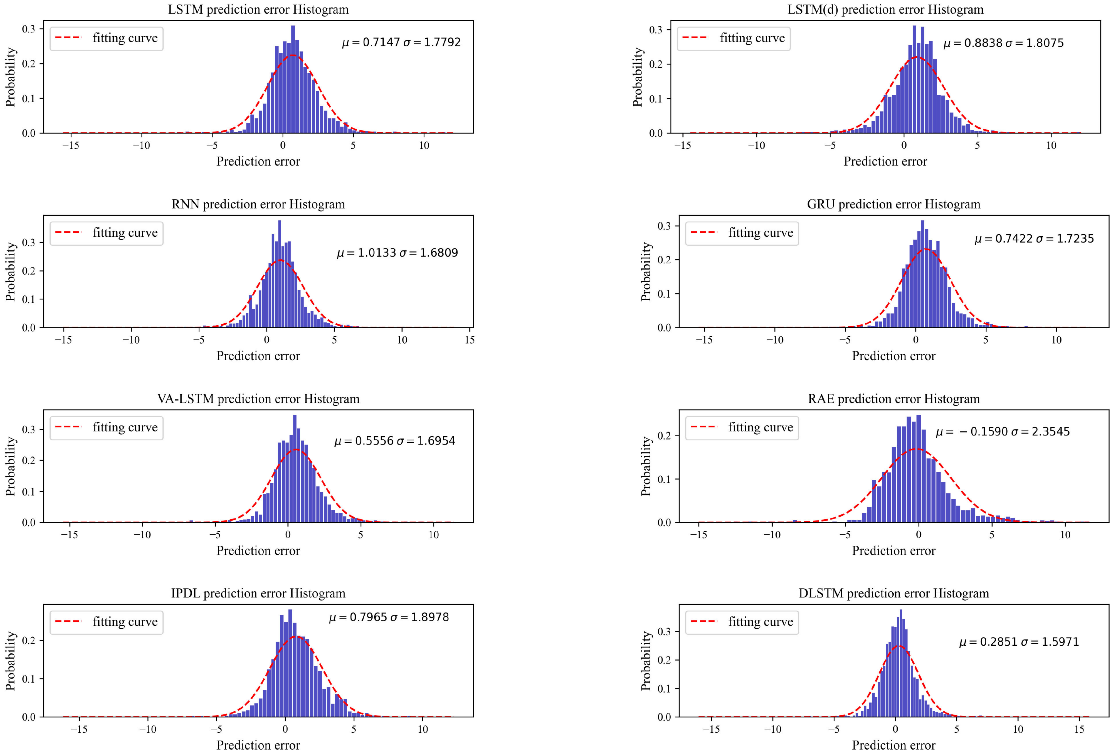


Fig. 8. Prediction error histogram of all models.

not compared with other models. The training time of DLSTM is just more than RNN, but less than other models. Meanwhile, the hidden variable of DLSTM is much bigger than the other models. The main reason may be that the difference module made the model converges fast, which makes a short training time. This also shows that the DLSTM model can achieve optimal performance quickly.

IV. Conclusion

In this article, to capture the impact of the difference information of the secondary variables on the key variables, a novel deep learning network called DLSTM was proposed for soft sensor modeling. The difference information of the original input, along with the current original input and pervious hidden cell state, was fed into the output gate to learn the actual dynamic characteristics of the industrial process. Then, the effectiveness of DLSTM was demonstrated by comparing with other soft sensor models based on deep learning in the case study on the grinding classification process. Besides, as the number of the DLSTM units increased, although there were more parameters need to be trained, the training time of DLSTM was less than most of other models. It was shown that the DLSTM model can achieve optimal performance quickly.

For future work, first, the DLSTM model can be trained in a semisupervised manner since the long sampling interval of key variables in actual complex industrial process. Then, when the working conditions change, the trained network may no longer adapt to the system. So it was important to explore an online learning method for DLSTM that the model can be periodically updated. At last, more knowledge of process mechanism should be combined to build a soft sensor model.

References

[1] M. Liukkonen, E. Hlikk, T. Hiltunen, and Y. Hiltunen, “Dynamic soft sensors for NOx emissions in a circulating fluidized bed boiler,” *Appl. Energy*, vol. 97, pp. 483–490, Sep. 2012.

[2] D. Wang, J. Liu, and R. Srinivasan, “Data-driven soft sensor approach for quality prediction in a refining process,” *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 11–17, Feb. 2010, doi: 10.1109/TII.2009.2025124.

[3] X. Wang and H. Liu, “Soft sensor based on stacked auto-encoder deep neural network for air preheater rotor deformation prediction,” *Adv. Eng. Informat.*, vol. 36, pp. 112–119, Apr. 2018, doi: 10.1109/TED.2016.2628402.

[4] Y. D. Ko and H. Shang, “A neural network-based soft-sensor for particle size distribution using image analysis,” *Powder Technol.*, vol. 212, no. 2, pp. 359–366, Oct. 2011.

[5] J. Yu, “Multiway Gaussian mixture model based adaptive kernel partial least squares regression method for soft sensor estimation and reliable quality prediction of nonlinear multiphase batch processes,” *Ind. Eng. Chem. Res.*, vol. 51, no. 40, pp. 13227–13237, Sep. 2012.

[6] A. Perera, N. Papamichail, N. Barsan, U. Weimar, and S. Marco, “On-line novelty detection by recursive dynamic principal component analysis and gas sensor arrays under drift conditions,” *IEEE Sensors J.*, vol. 6, no. 3, pp. 770–783, Jun. 2006.

[7] J. C. Gonzaga, L. A. Meleiro, C. Kiang, and R. M. Filho, “ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process,” *Comput. Chem. Eng.*, vol. 33, no. 1, pp. 43–49, Jan. 2009.

[8] Z. Ge, “Mixture Bayesian regularization of PCR model and soft sensing application,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4336–4343, Jul. 2015.

[9] J. Tang, T. Y. Chai, W. Yu, and L. J. Zhao, “Modeling load parameters of ball mill in grinding process based on selective ensemble multi-sensor information,” *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 726–740, Jul. 2013.

[10] V. Y. Kreinovich, "Arbitrary nonlinearity is sufficient to represent all functions by neural networks: A theorem," *Neural Netw.*, vol. 4, no. 3, pp. 381–383, 1991.

[11] A. K. Pani, V. K. Vadlamudi, and H. K. Mohanta, "Development and comparison of neural network based soft sensors for online estimation of cement clinker quality," *ISA Trans.*, vol. 52, no. 1, pp. 19–29, Jan. 2013.

[12] Y. L. He, Z. Q. Geng, and Q. X. Zhu, "Data driven soft sensor development for complex chemical processes using extreme learning machine," *Chem. Eng. Res. Des.*, vol. 102, pp. 1–11, Oct. 2015.

[13] Z. Ge and Z. Song, "A comparative study of just-in-time-learning based methods for online soft sensor modeling," *Chemometr. Intell. Lab. Syst.*, vol. 104, no. 2, pp. 306–317, Dec. 2010.

[14] X. Yuan, Z. Ge, B. Huang, and Z. Song, "A probabilistic just-in-time learning framework for soft sensor development with missing data," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 1124–1132, May 2017.

[15] C. Shang, F. Yang, D. Huang, and W. Lyu, "Data-driven soft sensor development based on deep learning technique," *J. Process Control*, vol. 24, no. 3, pp. 223–233, Mar. 2014.

[16] W. Yan, D. Tang, and Y. Lin, "A data-driven soft sensor modeling method based on deep learning and its application," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4237–4245, May 2017.

[17] L. Yao and Z. Ge, "Deep learning of semi-supervised process data with hierarchical extreme learning machine and soft sensor application," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1490–1498, Feb. 2018.

[18] X. Yuan, B. Huang, Y. Wang, C. Yang, and W. Gui, "Deep learning based feature representation and its application for soft sensor modeling with variable-wise weighted SAE," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3235–3243, Jul. 2018.

[19] X. Yuan, C. Ou, Y. Wang, C. Yang, and W. Gui, "Deep quality related feature extraction for soft sensing modeling: A deep learning approach with hybrid VW-SAE," *Neurocomputing*, vol. 396, pp. 375–382, Jul. 2020.

[20] X. Yuan, J. Zhou, B. Huang, Y. Wang, C. Yang, and W. Gui, "Hierarchical quality-relevant feature representation for soft sensor modeling: A novel deep learning strategy," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3721–3730, Jun. 2020.

[21] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2770–2779, Dec. 2017.

[22] M. Khodayar, J. Wang, and M. Manthouri, "Interval deep generative neural network for wind speed forecasting," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3974–3989, Jul. 2019.

[23] N. Zheng and J. Ding, "Regression GAN based prediction for physical of total hydrogen in crude oil," *Acta Auton. Sinica*, vol. 44, no. 5, pp. 915–921, May 2018.

[24] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.

[25] C. Chou *et al.*, "Physically consistent soft-sensor development using sequence-to-sequence neural networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2829–2838, Apr. 2020, doi: 10.1109/TII.2019.2952429.

[26] Q. Sun and Z. Ge, "Probabilistic sequential network for deep learning of complex process data and soft sensor application," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2700–2709, May 2019, doi: 10.1109/TII.2018.2869899.

[27] X. Yuan, L. Li, and Y. Wang, "Nonlinear dynamic soft sensor modeling with supervised long short-term memory network," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3168–3176, May 2020, doi: 10.1109/TII.2019.2902129.

[28] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic individual load forecasting using pinball loss guided LSTM," *Appl. Energy*, vol. 235, pp. 10–20, Feb. 2019.

[29] S. Wen, Y. Wang, Y. Tang, Y. Xu, P. Li, and T. Zhao, "Real-time identification of power fluctuations based on LSTM recurrent neural network: A case study on Singapore power system," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5266–5275, Sep. 2019, doi: 10.1109/TII.2019.2910416.

[30] X. Yuan, L. Li, Y. A. W. Shardt, Y. Wang, and C. Yang, "Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development," *IEEE. Trans. Ind. Electron.*, vol. 68, no. 5, pp. 4404–4414, May 2021.

[31] X. Yuan, L. Li, Y. Wang, C. Yang, and W. Gui, "Deep learning for quality prediction of nonlinear dynamic processes with variable attention-based long short-term memory network," *Can. J. Chem. Eng.*, vol. 98, no. 6, pp. 1377–1389, 2020.

[32] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[33] J. Heaton, *Introduction to Neural Networks With Java*. MO, Chesterfield: Heaton Research, Inc., 2008, pp. 158–159.

[34] Y. Wang, X. Chen, W. Gui, C. Yang, L. Caccetta, and H. Xu, "A hybrid multiobjective differential evolution algorithm and its application to the optimization of grinding and classification," *J. Appl. Math*, vol. 2013, pp. 1–15, Nov. 2013, doi: 10.1155/2013/841780.

[35] W. Dai, Q. Liu, and T. Chai, "Particle size estimate of grinding processes using random vector functional link networks with improved robustness," *Neurocomputing*, vol. 169, pp. 361–372, Dec. 2015.

Jiayi Zhou received the B.Eng. degree in automation and the M.Eng. degree in control science and engineering from Chongqing University, Chongqing, China, in 2014 and 2017, respectively. She is currently working toward the Ph.D. degree in control science and engineering with the School of Automation, Central South University, Changsha, China.

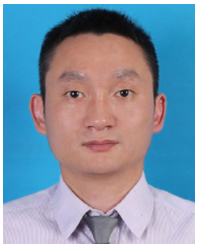
Her current research interests include modeling and optimal control of the complex industrial processes and soft sensor modeling based on data-driven.

Xiaoli Wang received the Ph.D. degree in control theory and control engineering from Central South University, Changsha, China, in 2011.

She was a Visiting Scholar with the Department of Energy Institute, Texas A&M University, TX, USA, from December 2016 to December 2017. She is currently a Full Professor with the School of Automation, Central South University. Her research interests include modeling and optimal control of the complex industrial processes, machine learning and pattern recognition, industrial process soft sensor modeling, process data analysis, etc.

Chunhua Yang (Senior Member, IEEE) received the M.Eng. degree in automatic control engineering and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 1988 and 2002, respectively.

She was with the Department of Electrical Engineering, Katholieke Universiteit Leuven, Leuven, Belgium, from 1999 to 2001. She is currently a Full Professor with the School of Automation, Central South University. Her research interests include modeling and optimal control of the complex industrial processes, intelligent control system, and fault-tolerant computing of real-time systems.



Wei Xiong received the B.Eng. degree in industrial electrical automation from Central South University, Changsha, China, in 2001.

He is currently the General Manager of new energy recycling and utilization department in Changsha Research Institute of Mining and Metallurgy Company Ltd and the Deputy Director of Hunan Engineering Technology Research Center for Echelon Utilization of Retired Battery. He is committed to the technical research and product development of the mineral processing control and Lithium-battery energy storage and echelon utilization, and realizing commercialized promotion.