IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 March 2023, accepted 1 April 2023, date of publication 12 April 2023, date of current version 18 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3266518

RESEARCH ARTICLE

A Dynamic Time Warping Based Locally Weighted LSTM Modeling for Temperature Prediction of Recycled Aluminum Smelting

YANHUI DUAN, JIAYANG DAI, YASONG LUO, GUANYUAN CHEN, AND XINCHEN CAI

Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, School of Electrical Engineering, Guangxi University, Nanning 530004, China

Corresponding author: Jiayang Dai (daijiayang@gxu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62273111.

ABSTRACT In the process of recycled aluminum smelting, timely measurement of the temperature of the smelting furnace is very important for the aluminum yield and quality. However, it is sometimes difficult or costly to measure the temperature in a timely manner due to the high temperature and pressure environment in the furnace. To tackle this problem, a soft sensor modeling framework which combines an operating condition classification and a prediction model based on locally sample-weighted long short-term memory (LSTM) neural network is proposed. In the operating condition classification, a hybrid of dynamic time warping (DTW) based fuzzy c-means and convolutional neural network is used to cluster the training samples and to classify the query samples. In the prediction model, the dynamic time warping and locally sample-weighted technique are introduced to LSTM to solve time-varying and strong nonlinear problems of the process. By adopting the method of classifying the operating conditions of the query samples before temperature prediction, the prediction time can be effectively reduced and the prediction accuracy can be maintained. The results of the experiment show that the proposed method can meet the prediction accuracy and time efficiency requirements of the regenerative aluminum smelting furnace.

INDEX TERMS Temperature prediction, just-in-time learning, dynamic time warping, fuzzy c-means, long short-term memory neural network.

I. INTRODUCTION

Aluminum and aluminum alloys are among the world's most widely used and economical metals due to their excellent mechanical properties, superior casting performance and high reserves. Nowadays, aluminum and its alloys play an irreplaceable role in aerospace, automotive manufacturing, and our everyday life.

Currently, aluminum can be obtained in two main ways, one by way of processing and smelting bauxite ore (primary aluminum), and the other by way of smelting sorted recycled aluminum (recycled aluminum). Compared to the production of primary aluminum, the production process of recycled aluminum is advantageous in terms of low energy consumption and low pollution. The production process of recycled aluminum includes the pretreatment of aluminum scrap, raw material sorting, recycled aluminum smelting, and refining, casting and so on. Among the above processes, aluminum smelting process is the most important part of recycled aluminum production, which determines the yield and quality of aluminum production and the energy consumption in the production process. The regenerative aluminum smelting furnace is being used on a large scale in the production of recycled aluminum because of its additional heat recovery device compared to the traditional recuperative aluminum smelting furnace, which can significantly improve the energy utilization and reduce the cost of the aluminum smelting process as well as the emission of exhaust [1].

The complex structure of the smelting furnace and the complicated smelting process make the aluminum smelting process a complex industrial process with typical features such as multi-variable, nonlinear, time-varying and large hysteresis.

The associate editor coordinating the review of this manuscript and approving it for publication was Orazio Gambino.

To establish the model of the complex industrial processes, numerous relevant studies have been made by many scholars in academia and industry. According to the principles of modeling, there are two modeling approaches: the mechanism modeling and the data-driven modeling [2]. The traditional process of mathematical modeling by mechanisms based on physical and chemical is called mechanism modeling. The mechanism modeling method requires a detailed analysis and simulation of the structure and production processes of industrial production machines [3], [4], [5], [6]. Therefore, it is very computationally intensive and time-consuming to model complex industrial processes using mechanism modeling methods. In addition, when performing mechanism modeling methods, some ideal situations are assumed for the modeled object, which can introduce unavoidable errors [7]. Data-driven modeling does not require much understanding of the internal structure and the operating principles of the modeled object, but only needs to understand the basic principles and identify its inputs and outputs. In recent years, data-driven soft sensors have increasingly been used to predict difficult-to-measure variables in complex industrial processes. Typical methods for building data-driven soft sensors include partial least squares (PLS) [8], [9], [10] and principal component regression (PCR) [11], [12], etc. However, the aforementioned soft sensor modeling methods are built by using a global, offline approach, and the models are difficult to update once they are built, which is detrimental to the prediction accuracy of soft sensors for time-varying complex industrial processes. To address these problems and achieve online modeling and updating, strategies of moving window (MV) [13] and just-in-time learning (JITL) [14] have been proposed. The strategy used by JITL builds models related to the entered query samples. In the JITL strategy, when a query sample is an input, a local model corresponding to the query sample is built by finding a certain number of samples with the highest similarity in the historical dataset according to a certain similarity measure. When a new query sample arrives, the old model is discarded and the new local model will be built based on the new query sample. Owing to its capability for online local modeling, JITL is well suited for dealing with nonlinear and time-varying problems in complex industrial processes [15]. For example, Dai et al. [7] proposed a data-driven soft sensor model based on a combination of the moving window technique and the JITL strategy, and validated the effectiveness of the hybrid model by combining this data-driven model with the mechanistic model on the kiln. Zhang et al. [16] used a moving window strategy to update the historical database and a soft sensor based on a JITL strategy and a regularized limit learning machine (RELM) to predict the burning zone temperature of the rotary kiln sintering process, and achieved a more accurate prediction result.

Artificial intelligence neural networks (ANNs) have been proven to be one of the most promising methods for soft sensor modeling of complex industrial processes due to their powerful ability of handling nonlinear processes [17], [18]. However, the general ANN is a static structure, which makes it difficult to extract the time dynamic information of complex industrial processes. For recurrent neural network (RNN) in ANNs, each of their hidden layer nodes forms a loop, allowing the output of hidden layer nodes to influence the subsequent inputs of the same nodes, which gives RNN the ability to remember historical information [19]. The recurrent structure allows RNN to exhibit temporal dynamics, which is a significant advantage in processing industrial process data with significant temporal characteristics [20]. However, since standard RNN may suffer from gradient explosion or gradient disappearance when dealing with long time series [21], long short-term memory neural network (LSTM) was proposed by Hochreiter and Schmidhuber to solve the aforementioned problems that are present in RNN [22]. The three-gate structure of the memory gate, forgetting gate, and output gate adopted by LSTM, especially the forgetting gate which can selectively forget and discard some past information, allows LSTM to effectively control the convergence of the gradient during training. Meanwhile, the problem of unaccepted gradient disappearance and gradient explosion are greatly alleviated by the structure of LSTM, which allows LSTM to be more effectively applied to data-driven modeling in complex industrial processes than RNN.

For the JITL strategy, the accuracy of the model depends on the association between the selected training samples to be modeled locally and the query samples. For the standard JITL strategy, the training samples are fed directly into the model for training, which causes the model to lose accuracy in predicting the output values of the query samples. Therefore, it is necessary to consider the similarity between the selected training samples and the query sample before training the JITL model. The JITL strategy based on locally sample-weighted considers the degree of similarity between the query sample and each training sample and uses this degree of similarity as the weight of the corresponding training sample, which greatly improves the JITL model's ability to handle nonlinearities. The traditional approach of JITL uses the Euclidean distance as the similarity measure between the historical and query samples, and the samples usually contain only one sampling point [11], [14], [23], which is unfavorable for complex industrial processes with time-series characteristics. When the sliding window method is used to obtain samples with a certain time length, the original temporal characteristics of the data are effectively retained. As an algorithm that can measure the similarity between temporal segments, dynamic time warping (DTW) [24], [25] is better when applied to measure the similarity between samples after the sliding window. Based on the above problems, a DTW-based locally sample-weighted LSTM model (DLWLSTM) is proposed in this paper, which uses the DTW distance as an index of the distance between historical and query samples in sample weighting and can better extract the nonlinear features related to the output variables.

While the JITL strategy improves the prediction accuracy by coping with the nonlinearity and time-varying of the modeled objects by building an corresponding model for each query sample when performing local modeling, which also increases the time consumption on predicting query samples. Strategies for the selective updating of models have been proposed to reduce the time consumption on modeling. Chen et al. [11] proposed a local modeling model update strategy based on approximate linearity dependence (ALD), which determines whether to update the local model by calculating the ALD value between the query sample and the training sample of the previous local model, and this local model selective update strategy maintains the accuracy of the prediction while reducing the time used for prediction. In this study, an operating condition classification and prediction model consisting of a DTW-based fuzzy c-means (FCM) algorithm [26] and convolutional neural network (CNN), which is denoted as DFC, is proposed. The idea of classifying data before modeling can be realized by combining DFC and DLWLSTM models (DFC-DLWLSTM). In the DFC-DLWLSTM model, each query sample only needs to find local modeling samples in the historical sample database of the category to which the query sample belongs, which significantly reduces the time required to find local modeling samples for query samples. At the same time, the relationship between the mean clustering center of the category after clustering and the query sample is used as an index for training sample selection in local modeling, which ensures that the modeling time is significantly reduced while the prediction accuracy is maintained.

The main structure of this paper is as follows: The next section shows the process of aluminum smelting in regenerative aluminum smelting furnaces, and analyzes the problems and modeling difficulties in the aluminum smelting process. In the third part, the DFC-DLWLSTM-based soft sensor model proposed in this paper is introduced. In the fourth part, the proposed model is applied to the data collected from a regenerative aluminum smelting plant to validate its effectiveness. Finally, the conclusions of the full paper will be made.

II. PROBLEM ANALYSIS OF ALUMINUM SMELTING PROCESS

Aluminum smelting process is a complex thermodynamic process, and the procedure for the recycled aluminum smelting process is shown in Figure 1. Throughout the aluminum smelting process, there is a dynamic change in the temperature of the furnace chamber and liquid aluminum as heat flows in and out. The temperature of the aluminum smelting furnace chamber (furnace temperature) is an important control parameter for the aluminum smelting process. The furnace temperature controls the temperature of the liquid aluminum in the smelting furnace, and the accuracy of the furnace temperature control determines the yield and quality of the aluminum. Therefore, it is extremely important to accurately and quickly sense the dynamic changes in the furnace temperature during the aluminum smelting process.

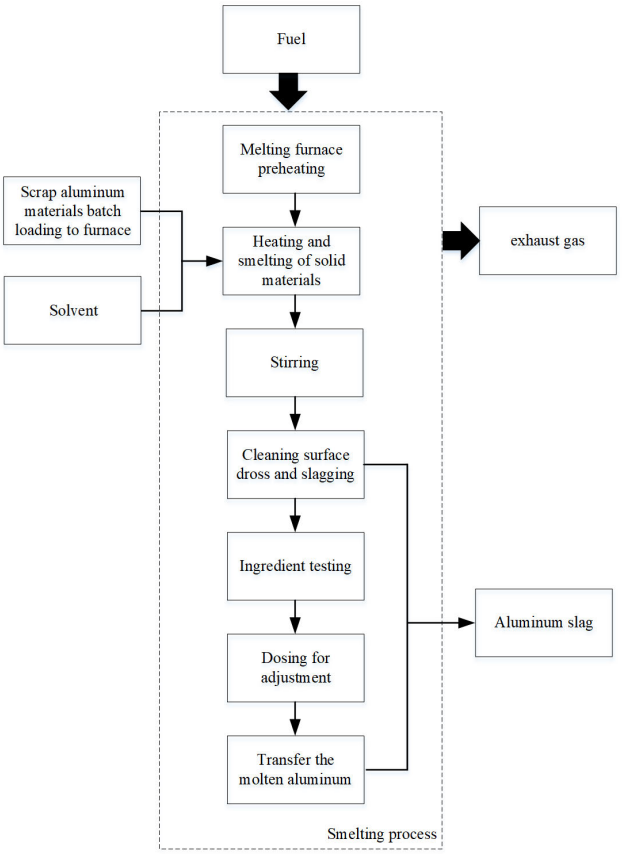


FIGURE 1. Flow chart of aluminum smelting process.

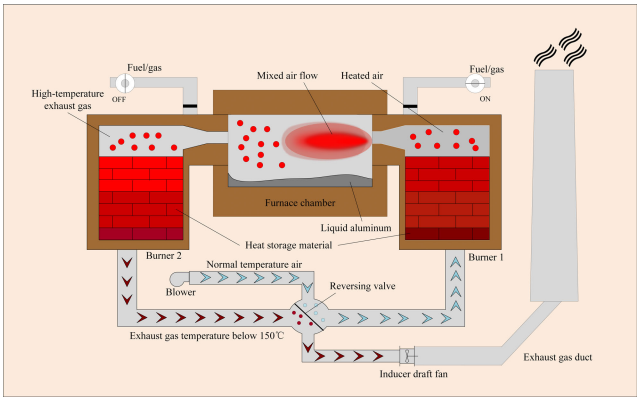


FIGURE 2. Structure and working principle of regenerative aluminum smelting furnace.

The regenerative smelting furnace system is a typical complex industrial process control system. Figure 2 shows the structure and working principles of a regenerative aluminum smelting furnace. The regenerative aluminum smelting furnace is mainly composed of a furnace chamber, regenerative burner (including regenerative chamber and burner), air/fume duct, reversing device, and fume exhaust device. As shown in

FIGURE 3. Pictures of furnace door and liquid aluminum temperature thermocouple.

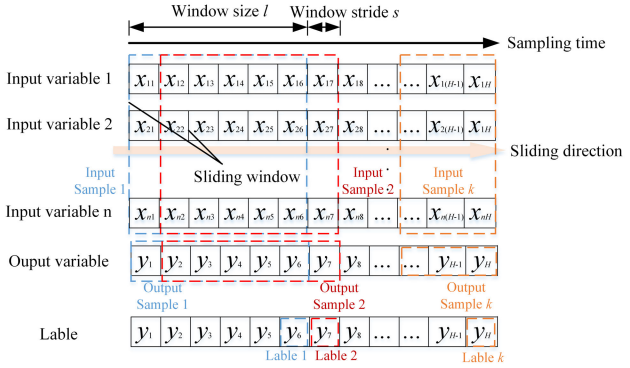


FIGURE 4. Sliding window working schematic.

Figure 2, the heat-regenerative burners of a heat-regenerative aluminum smelting furnace are arranged in pairs and symmetrically in the furnace structure, and a heat-regenerative aluminum smelting furnace generally has one or more pairs of heat-regenerative burners. For a pair of heat regenerative burners called Burner 1 and Burner 2, they are not in operation simultaneously. When the regenerative burner Burner 1 is in operating condition, the fuel path of Burner 1 is opened and the reversing valve is in such a state that the air/fume duct of Burner 1 is fed with air, which is blown in by the blower, flowed through the reversing valve and then heated rapidly to 80%-90% of the furnace chamber temperature by the regenerative chamber of Burner 1 before entering the furnace chamber through the burner of Burner 1. The heated high temperature air will be mixed with the fumes in the furnace chamber after entering the furnace chamber, forming a high temperature oxygen-poor air flow which is much lower than the normal air oxygen content of 21%; fuel is then injected into the oxygen-poor high temperature air, and the fuel will be combusted in the oxygen-poor state. The other heat regenerative burner Burner 2 is now in the exhaust and heat storage state, and the fuel path of Burner 2 is closed. The high temperature flue gas from the furnace chamber will pass through the thermal chamber of Burner 2 and heat the regenerator, which prepares the Burner 2 to heat the air entering the furnace chamber when Burner 2 is in operation. After passing through the heat storage chamber, the fume will be directly exhausted into the atmosphere through the reversing valve, and the temperature of the fume exhausted into the atmosphere is generally less than 150 °C. When the reversing valve changes state, the state of Burner 1 and Burner 2 will switch correspondingly.

The temperature of the liquid aluminum is measured using a thermocouple on the smelting furnace. When the furnace door is closed, the thermocouple is inserted into the liquid aluminum for measurement as required. When the furnace door is opened, generally because of slagging, the thermocouple is automatically withdrawn to prevent damages from the slagging equipment. Figure 3 shows the door and the thermocouple of smelting furnace.

Since the measurement of furnace temperature is simpler and less expensive than the measurement of aluminum liquid temperature, the measurement of furnace temperature is generally used to guide the production process in the aluminum smelting industry. However, the furnace temperature has a lag in the control of the liquid aluminum temperature, so it is necessary to predict the change of the furnace temperature in advance according to the relevant variables of the aluminum smelting furnace.

III. THE PROPOSED DFC-DLWLSTM

A. DATA PRE-PROCESSING WITH SLIDING WINDOW

When collecting raw industrial data, the data collected at each sampling point are the values of each variable at a certain moment. However, for complex industrial processes, the value of a certain output variable at a given moment does not depend on the input at that moment alone, but also on the changes in the input at several previous moments. Therefore, it is necessary to use the input at several moments before the sampling moment to map the output at the sampling moment together.

In order to eliminate the influence of magnitude between variables, data standardization is required to address the comparability between data. After the raw data are processed by data standardization, the variables have the same order of magnitude and are suitable for comparison and evaluation. The most typical data standardization method is data normalization. In this study, z-score standardization is used, which is more applicable to the method proposed in this paper, the formula of z-score is shown in Equation (1):

$$x^* = \frac{x - \mu}{\sigma_z}, \tag{1}$$

where μ and σ_z denote the mean and variance of the raw data, respectively, and x^* is the normalized value. Supposing we have the normalized raw data sample set (X, Y), where $X = \{[x_{1h}, x_{2h}, \ldots\ldots, x_{nh}]\}_{h=1}^{H}$ is the input sample set, $Y = \{y_h\}_{h=1}^{H}$ is the output sample set, and n is the number of input variables, H is the number of raw data sampling points. Now using a sliding window with window size l and stride s to

slide the raw data, and we will get:

$$X^L = \{X_w^L\}_{w=1}^W, \tag{2}$$
$$Y^L = \{Y_w^L\}_{w=1}^W, \tag{3}$$

in which

$$X_w^L = \{X_{wh}^L\}_{h=ws}^{ws+l-1} = \{[x_{1h}, x_{2h}, \ldots\ldots, x_{nh}]\}_{h=ws}^{ws+l-1} \tag{4}$$

is the input sample,

$$Y_w^L = \{Y_{wh}^L\}_{h=ws}^{ws+l-1} = \{y_h\}_{h=ws}^{ws+l-1} \tag{5}$$

is the output sample, and $W = \lfloor (H-l)/s \rfloor + 1$ is the number of samples after the sliding window. For the w-th input sample X_w^L and output sample Y_w^L after the sliding window, let

$$[x_{(w)1u}, x_{(w)2u}, \ldots\ldots, x_{(w)nu}] = [x_{1h}, x_{2h}, \ldots\ldots, x_{nh}], \tag{6}$$
$$y_{(w)u} = y_h, \tag{7}$$

where $(u, h) = (1, ws), (2, ws+1), \ldots\ldots, (l, ws+l-1)$, then we will have

$$X_w^L = \{X_{wu}^L\}_{u=1}^l = \{[x_{(w)1u}, x_{(w)2u}, \ldots\ldots, x_{(w)nu}]\}_{u=1}^l, \tag{8}$$
$$Y_w^L = \{Y_{wu}^L\}_{u=1}^l = \{y_{(w)u}\}_{u=1}^l. \tag{9}$$

The value Y_w^L of the last s sampling points of Y_w^{-s} is taken as the label of the input variable after the sliding window, that is

$$Y_w^{-S} = \{y_{(w)u}\}_{u=l-s}^l = \{y_h\}_{h=ws+l-s}^{ws+l-1}. \tag{10}$$

The sample set obtained after the sliding window is (X^L, Y^{-S}).

B. METHOD OF OPERATING CONDITION CLASSIFICATION AND PREDICTION BASED ON DFC

For the aluminum smelting process, the temperature trend varies depending on the smelting stage. When building soft sensors, if the data collected from the aluminum smelting process can be clustered according to the operating conditions in advance, the total time required for soft sensor modeling and prediction can be effectively reduced while maintaining the prediction accuracy of the soft sensor.

There are several input variables of the aluminum smelting process, and it is complicated to cluster multivariate time series directly. Therefore, this paper explores a classification method combining the unsupervised clustering method and supervised classification method to achieve the classification of multivariate time series. First, the unsupervised clustering algorithm DTW-FCM is applied to cluster the output variable samples of training samples and then the clustered output variable samples will be labeled with categories. Subsequently, the input variable samples corresponding to the output variable samples and the corresponding category labels will be formed as a training sample set to train the CNN to obtain a classification model. Finally, a classification method that can predict the operating conditions of the query samples, i.e., supervised classification method, will be obtained.

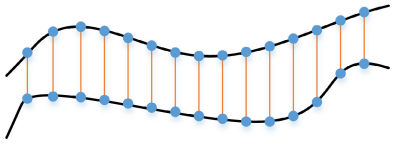


FIGURE 5. One-to-one correspondence of time series points to points.

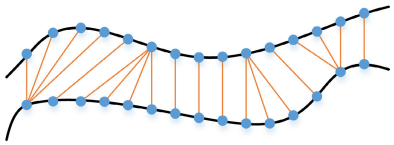


FIGURE 6. Example of point-to-point correspondence after DTW processing.

1) LABELING OUTPUT VARIABLE SAMPLES WITH CATEGORIES BY METHOD OF DTW-FCM CLUSTERING

The FCM algorithm is an algorithm based on fuzzy partitioning, whose basic idea of partitioning categories is to maximize the differences between clusters and minimize the differences within clusters. Supposing $x = \{x_j\}_{j=1}^g$ is the data set to be clustered into c clusters by FCM, where $x_j = \{x_{1j}, x_{2j}, \ldots, x_{nj}\}$, and n is the number of variables of that data set. For the i-th cluster, the degree of membership of the j-th point to that cluster is given by the membership u_{ij}, which takes values in the range [0, 1] and meet the constraint $\sum_{i=1}^c u_{ij} = 1$. The closer u_{ij} is to 1, the more similar the sample point is to the cluster it is clustered in, i.e., the higher the degree of membership to that cluster. And on the contrary, the lower the degree of membership.

The objective function of the FCM algorithm is

$$J_m(U, C) = \sum_{i=1}^{c} \sum_{j=1}^{g} u_{ij}^m d_{ij}^2(x_j, c_i). \tag{11}$$

In the above objective function, $d_{ij}(x_j, c_i)$ denotes the clustering index, and the clustering index in the traditional FCM algorithm is the Euclidean distance between the sample point x_j and the clustering center c_i. m ($m > 1$) is the fuzzy weight index, which portrays the fuzziness of the classification. If m is too large, the clustering effect will be poor because of the great fuzziness in clustering, while if m is too small the algorithm will be close to the k-means clustering algorithm. The fuzzy partition matrix U called the membership matrix is an matrix of size $g * c$, which is consisted of the membership degree between each sample and each cluster. C is the matrix composed of the clustering centers c_i of each cluster.

When the traditional FCM algorithm uses the Euclidean distance to compare the similarity of two time series of the same length, it compares the points of the time series in a fixed order and cannot consider the existing time shift, that is, a one-to-one correspondence, as shown in Figure 5.

The DTW algorithm automatically warps the time series to make the shape of the two series as identical as possible and to obtain the maximum possible similarity, as shown in Figure 6.

DTW can better portray the similarity between two time series based on their shapes, so using the DTW distance as the clustering index when clustering the time series with the FCM algorithm will have better results.

Assuming that $x = \{x_i\}_{i=1}^{l_x}$ and $y = \{y_j\}_{j=1}^{l_y}$ are time series, and l_x and l_y denote the length of the two time series respectively. An $l_x * l_y$ matrix D is developed by a defining distance of each point between the series, called distance matrix. Denoting a possible correspondence between x and y by ϕ, then $\phi(q) = (i, j)_q$ $(q = 1, 2, \ldots, Q)$ denotes the q-th correspondence corresponding to the i-th element of x and the j-th element of y, where Q is the product of the length of time series x and y. The DTW algorithm is to find an optimal path from the upper right corner element to the lower left corner element in the distance matrix D, which corresponds to an optimal corresponding series ϕ'. The sum of the elements on the ϕ' path is the smallest among the elements of any path from the upper right corner to the lower left corner of the D matrix, and this minimum value is the minimum accumulated distortion value. Suppose the number of elements on the diagonal of the distance matrix D is K, then the minimum accumulated distortion value can be calculated as:

$$Dist_{\phi'}(x, y) = \sum_{k=1}^{K} E((i, j)_k), \tag{12}$$

where $E((i, j)_k)$ represents the distance of the k-th correspondence, i.e., the distance of element x_i and y_j, and the Euclidean distance is usually used to calculate that distance.

When the DTW distance is used as the clustering index of FCM, the corresponding objective function of FCM is

$$J_m(U, C) = \sum_{i=1}^{c} \sum_{j=1}^{g} u_{ij}^m Dist_{\phi'.ij}^2(x_j^L, c_i^L), \tag{13}$$

where $Dist_{\phi'.ij}(x_j^L, c_i^L)$ is the DTW distance between the time series x_j^L and the i-th cluster center c_i^L, and x_j^L is the time series of length l corresponding to the j-th sample point x_j. When the objective function J_m takes the minimum value, the calculation formula of the cluster center is

$$c_i^L = \frac{\sum_{j=1}^{g} u_{ij}^m x_j^L}{\sum_{j=1}^{g} u_{ij}^m}, \tag{14}$$

and the membership degree of the j-th time series to the i-th category is

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{Dist_{\phi'.ij}(x_j^L, c_i^L)}{Dist_{\phi'.kj}(x_j^L, c_k^L)}\right)^{\frac{2}{m-1}}}. \tag{15}$$

The DTW-FCM algorithm operates as follows:

Step 1: Initialize the membership matrix U by random numbers.

Step 2: Calculate the clustering center of each cluster by Equation (14).

Step 3: Calculate the DTW distance between each sample and each the cluster center.

Step 4: Update each element of the membership matrix u_{ij} by Equation (15).

Step 5: Calculate the value of the objective function by Equation (13) and return to Step 2 if the value of the objective function is greater than the set threshold. If the value of the objective function is less than the threshold or the number of iterations reaches the set value, the algorithm ends its run and outputs the final membership matrix U.

The category to which each clustered sample belongs can be determined by the membership matrix U. By clustering the output variable samples after the data sliding window, the category to which each output variable sample belongs can be determined, which also corresponds to the category of the input variables as well, i.e., the clustering of the training sample set is completed. The categories of input variable samples are used as labels for the corresponding input variable samples to form a training set (C^L, V), which will be used to train the CNN. In the training set (C^L, V),

$$\begin{aligned} C^L &= \{C_i^L\}_{i=1}^{c} = \{\{C_{ik}^L\}_{k=1}^{K_i}\}_{i=1}^{c} \\ &= \{\{\{[x_{1h}, x_{2h}, \ldots\ldots, x_{nh}]\}_{h=ws}^{ws+l}\}_{k=1}^{K_i}\}_{i=1}^{c} \\ &= \{\{\{[x_{(k)1u}, x_{(k)2u}, \ldots\ldots, x_{(k)nu}]\}_{u=1}^{l}\}_{k=1}^{K_i}\}_{i=1}^{c} \end{aligned} \tag{16}$$

denotes the k-th sample in cluster i, where K_i denotes the number of samples in cluster i, and w denotes the w-th sample in the sample set obtained from the raw data processed by the sliding window. And

$$V = \{V_i^c\}_{i=1} = \{\{onehot(i)_j\}_{k=1}^{K_i}\}_{i=1}^{c}, \tag{17}$$

where $onehot(i)_j$ is the one-hot encoding with label i for the j-th sample in cluster i (for a certain cluster of samples, the labels are all the same), and the length of the one-hot encoding eigenvector is c.

For each cluster of input samples

$$C_i^L = \{\{[x_{(k)1u}, x_{(k)2u}, \ldots\ldots, x_{(k)nu}]\}_{u=1}^{l}\}_{k=1}^{K_i}, \tag{18}$$

the mean center of the cluster can be found by Equation (18):

$$\begin{aligned} \Lambda_i^L &= \{[x_{1u}^{\Lambda_i}, x_{2u}^{\Lambda_i}, \ldots\ldots, x_{nu}^{\Lambda_i}]\}_{u=1}^{l} \\ &= \left\{ \frac{\sum_{i=1}^{K_i} [x_{(k)1u}, x_{(k)2u}, \ldots, x_{(k)nu}]}{K_i} \right\}_{u=1}^{l}. \end{aligned} \tag{19}$$

2) THE PROPOSED DFC OPERATING CONDITION CLASSIFICATION METHOD

After clustering the training samples for operating conditions using the DTW-FCM algorithm, a sample set with labels that can be used for supervised learning is obtained. CNN uses the sliding of convolutional kernels to achieve a good extraction of local hidden features of the data, and CNNs are generally divided into 1D-CNN and 2D-CNN. The sliding direction of the convolution kernel of the 2D-CNN has two dimensions,

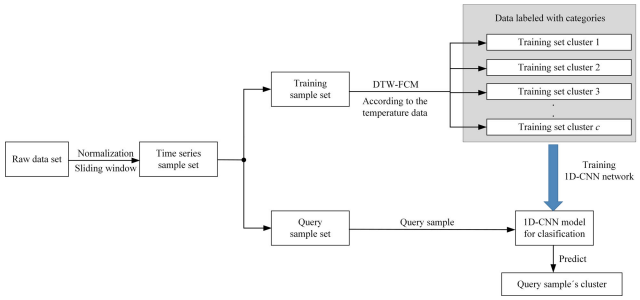


FIGURE 7. The flow chart of the DFC algorithm.

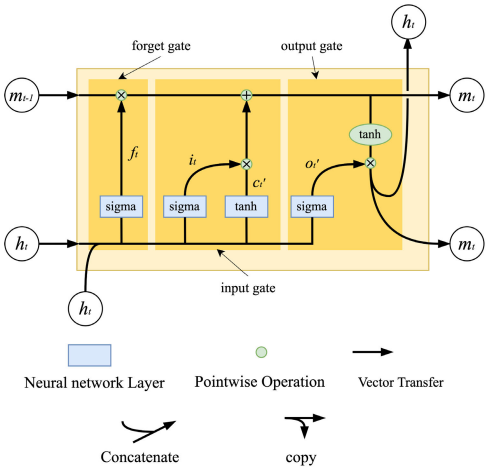


FIGURE 8. The basic cell unit of LSTM neural network.

which is advantageous for processing data that are associated in both dimensional directions, such as image data. However, for time-series data such as industrial production data, the advantage of 1D-CNN comes into play to better obtain the time-series information of the samples. The sliding direction of the convolution kernel in the 1D-CNN is only one dimension, which can be good for sliding from the time dimension to obtain the features of the temporal data.

The 1D-CNN used in this paper contains three Convolutional layers, two Max Pooling layers, an intermediate Full Connected Layer with a sigmoid activation function, and an output Fully Connected Layer with an activation function of softmax.

After training the 1D-CNN with category-labeled training sample set (C^L, V), the 1D-CNN can be used to predict the category of query samples. The flow chart of the DFC algorithm is shown in Figure 7.

C. SAMPLE-WEIGHTED LSTM BASED ON DTW DISTANCE

LSTM is a type of network improved from RNN, which have the capability to process long time series.Figure 8 shows the basic cell unit of LSTM neural network.

The three-gate structure of LSTM can more effectively solve the gradient disappearance and gradient explosion problems compared to RNN, so that LSTM have a great advantage in processing long time series [13]. The key to the LSTM is the state of its basic unit, m_t, as shown in Figure 8. m_t runs through the entire chain of LSTM transmission and that is the reason why LSTM networks can solve the problems in the processing of long time series data. The three stages within the LSTM are as follows.

1) FORGETTING STAGE (FORGETTING GATE)

This stage focuses on the selective forgetting of the input passed in from the previous node. Specifically, the calculated f_t is used as a forgetting gating to control what needs to be forgotten and left behind in the previous cell state m_{t-1}. f_t is calculated as

$$f_t = \text{sigma}(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \tag{20}$$

where $\text{sigma}(x) = 1/(1 + \text{e}^{-x})$.

2) SELECTIVE MEMORY STAGE (INPUT GATE)

This stage receives the input from this stage and remembers it selectively, i.e., it decides which new information will be stored in the next cell state m_t. In this stage, the sigmoid layer decides which values will be updated to obtain i_t, while the tanh layer creates a new vector c'_t of candidate values. The information obtained from the above two layers together with the cell state m_{t-1} from the previous node will determine the update of the cell state of that node. And i_t, c'_t and the cell state m_t can be calculated separately by following equations:

$$i_t = \text{sigma}(W_{ix}x_t + W_{ih}h_{t-1} + b_i), \tag{21}$$
$$c'_t = \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c), \tag{22}$$
$$m_t = f_t \odot m_{t-1} + i_t \odot c'_t, \tag{23}$$

where $\odot$ indicates a point-by-point operation, i.e., the corresponding elements are multiplied.

3) OUTPUT STAGE (OUTPUT GATE)

This stage will determine the output of the current hidden state h_t. First, a sigmoid layer is used to decide which parts of the input and the hidden state of the previous node will be output, i.e., to get o_t, then the cell state is processed by the tanh function to obtain a result, and finally the output, i.e., the current hidden state h_t is obtained by multiplying o_t with the result of the tanh function. o_t and the hidden state h_t are calculated as follows:

$$o_t = \text{sigma}(W_{ox}x_t + W_{oh}h_{t-1} + b_o), \tag{24}$$
$$h_t = o_t \odot \tanh(m_t), \tag{25}$$

By combining the advantages of LSTM, locally sample-weighted based JITL and the superiority of DTW distance in similarity measurement, a DTW-based locally sample-weighted LSTM soft sensor model is proposed in this paper. First, the sum of the DTW distance between each segment of the query sample and the corresponding segment of each historical sample is calculated as the DTW distance between the query sample and each historical sample. Then the N historical samples with the smallest DTW distance are

selected as the local modeling sample set, and the samples are assigned certain weights according to the corresponding DTW distance values. Finally, each sample is multiplied by the corresponding weights to form a new local modeling sample set.

Assuming that the query sample X_q^L belongs to cluster i, then the local modeling samples of X_q^L will be found from the historical sample set of cluster i. However, if the query sample is misclassified, the local model corresponding to the query sample will suffer a significant error. To improve the error tolerance when modeling query samples, using the DTW distance between the query sample and the mean cluster center Λ_i^L of each cluster of historical samples as a reference for the selection of query sample's local modeling sample is proposed. The DTW distance between the query sample and the mean cluster center of each cluster can be calculated as follows:

$$\begin{aligned} D_{\Lambda_i} &= \sum_{u=1}^{l} Dist_{\phi'}(\Lambda_{iu}^L, X_{qu}^L) \\ &= \sum_{u=1}^{l} Dist_{\phi'}([x_{1u}^{\Lambda_i}, x_{2u}^{\Lambda_i}, \ldots, x_{nu}^{\Lambda_i}], \\ &\quad [x_{(q)1u}, x_{(q)2u}, \ldots, x_{(q)nu}]), \\ &\quad i = 1, 2, \ldots, c \end{aligned} \tag{26}$$

where c is the number of clusters in the classification of operating conditions, and i, q represent the i-th mean cluster center and the q-th query sample respectively.

For a given query sample X_q^L, there are two possible cases for D_{Λ_i}:

Case 1: The category i corresponding to the smallest value of D_{Λ_i} is as the same as the category corresponding to X_q^L when it is classified.

Case 2: The category i corresponding to the smallest value of D_{Λ_i} is different from the category corresponding to X_q^L when it is classified.

For Case 1, the two methods are cross-validated to ensure that the classification of the query sample is correct. The historical sample set of the cluster to which X_q^L belongs at the time of classification can be used as the historical sample set for the local modeling of X_q^L. For Case 2, that situation means that the query sample X_q^L may be incorrectly classified by the DFC algorithm, and the historical sample set of X_q^L's local modeling needs not only the sample set of the category corresponding to the classified category, but also the sample set of the category corresponding to D_{Λ_i} with the smallest value at this time as a supplement. This is the strategy proposed in this paper for the selection of the historical sample set when local modeling.

The local modeling samples are selected by evaluating the distance and similarity between the query samples and each historical input sample, and different weights are assigned to each local modeling sample according to the similarity between the query samples and the selected local modeling samples. Supposing that after using the strategy for the selection of the historical sample set when local modeling, the input and output samples of the local modeling historical sample set for the query sample X_q^L are $X_H^L = \{X_w^L\}_{w=1}^{W_H}$ and $Y_H^{-S} = \{Y_w^{-S}\}_{w=1}^{W_Q}$ respectively, where W_Q is the number of the determined historical sample set. In this paper, the DTW distance is used to calculate the similarity between the q-th query sample X_w^L and each historical sample, and the similarity is defined as follows:

$$\begin{aligned} D_w &= \sum_{u=1}^{l} Dist_{\phi'}(X_{iu}^L, X_{qu}^L) \\ &= \sum_{u=1}^{l} Dist_{\phi'}([x_{(w)1u}, x_{(w)2u}, \ldots, x_{(w)nu}], \\ &\quad [x_{(q)1u}, x_{(q)2u}, \ldots, x_{(q)nu}]), \\ &\quad w = 1, 2, \ldots, W. \end{aligned} \tag{27}$$

where i, q represent the i-th historical sample and the q-th query sample respectively.

The designated weight of the sample X_w^L is calculated as

$$\Omega_w = \exp(-D_w^2/\sigma^2),\ w = 1, 2, \ldots, W \tag{28}$$

where σ is a parameter that adjusts the rate of change of sample weights at different similarity distances.

From Equation (28), it can be observed that the higher the similarity degree between the historical sample and the query sample is (i.e., the smaller the DTW distance between the historical sample and the query sample), the higher the weight of the historical sample under that query sample will be. After the weights $\{\Omega_w\}_{w=1}^{W}$ between all historical samples and query samples is obtained, arranging them in order from highest to the lowest and taking the first N value of the weight from the arranged weight set as the local modeling sample's weight set $\{\Omega_n\}_{n=1}^{N}$. The samples corresponding to the weights $\{\Omega_n\}_{n=1}^{N}$ will be the samples for local modeling, and the input and output samples of the local modeling training set can be obtained as $X_{local}^L = \{X_n^L\}_{n=1}^{N}$ and $Y_{local}^{-S} = \{Y_n^{-S}\}_{n=1}^{N}$ respectively. Then the input sample weighted by the sample weights can be expressed as

$$\begin{aligned} \widetilde{X}_{local}^L &= \{\Omega_n X_n^L\}_{n=1}^{N} = \{\{\Omega_n X_{nu}^L\}_{u=1}^{l}\}_{n=1}^{N} \\ &= \{\{[\Omega_n x_{(n)1u}, \Omega_n x_{(n)2u}, \ldots\ldots, \Omega_n x_{(n)nu}]\}_{u=1}^{l}\}_{n=1}^{N}, \end{aligned} \tag{29}$$

where n, u denote the u-th sample point of the n-th sample in the local modeling sample set respectively.

After obtaining the locally sample-weighted samples $\widetilde{X}_{local}^L$, the LSTM is trained with $\widetilde{X}_{local}^L$. For the u-th sample point of the n-th local modeling sample, the hidden layer state of the LSTM can be obtained by forward propagation as:

$$f_u = \text{sigma}(W_{fx}(\Omega_n X_{nu}^L) + W_{fh} h_{u-1} + b_f) \tag{30}$$

$$i_u = \text{sigma}(W_{ix}(\Omega_n X_{nu}^L) + W_{ih} h_{u-1} + b_i) \tag{31}$$

$$c_u' = \tanh(W_{cx}(\Omega_n X_{nu}^L) + W_{ch} h_{u-1} + b_c) \tag{32}$$

$$m_u = f_u \odot m_{u-1} + i_u \odot c_t' \tag{33}$$

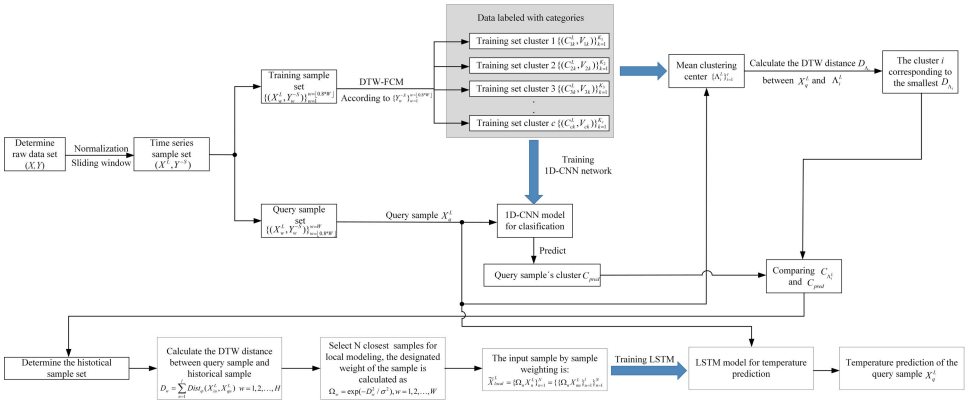


FIGURE 9. The flow chart of the proposed DFC-DLWLSTM method.

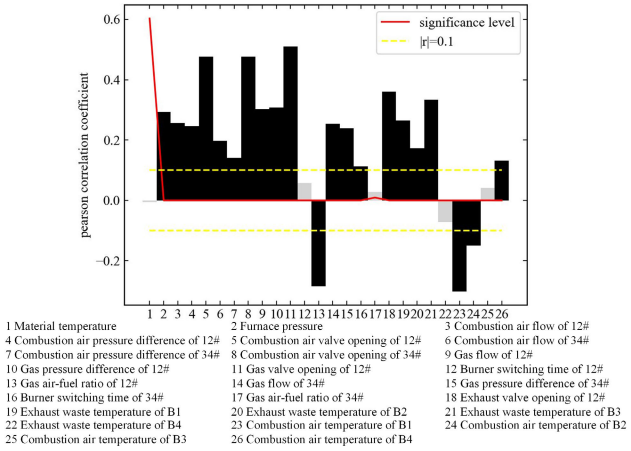
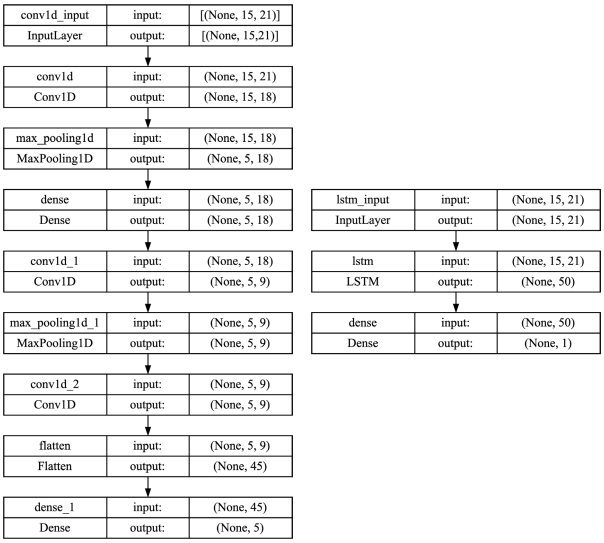


FIGURE 10. Name of each variable and its Pearson coefficient with furnace temperature.



(a) CNN model (b) LSTM model

FIGURE 11. The structure of CNN model and LSTM model used in this paper.

$$o_u = \text{sigma}(W_{ox}(\Omega_n X_{nu}^L), W_{oh} h_{u-1} + b_o) \quad (34)$$

$$h_u = o_u \odot \tanh(m_u) \quad (35)$$

The overall flow chart of the proposed DFC-DLWLSTM method is shown in the Figure 9.

IV. INDUSTRIAL APPLICATION

To verify the prediction accuracy of the soft sensor based on the proposed DFC-DLWLSTM model, the model is applied to the furnace temperature prediction of a regenerative aluminum smelting furnace. The data used in the model validation are a raw data set of 2021 sampling points, which were collected from the aluminum smelting process of a regenerative aluminum smelting plant from November 1st to 8th, 2017, with a sampling interval of 5 minutes. The number of variables in the raw data set is 30, and due to the sampling errors caused by the damaged sensors, three of them are removed. Among the remaining 27 variables, the furnace temperature is used as the output variable, and the other 26 variables are selected based on the magnitude of the Pearson coefficient between them and the furnace temperature, excluding those with an absolute value of Pearson coefficient less than 0.1 with the furnace temperature, and taking those with an absolute value of Pearson coefficient greater than 0.1 and significance level less than 0.05 with the furnace temperature as the input variable to the soft sensor model. The names of the input variables and the Pearson coefficients between them and the furnace temperature are shown in the Figure 10, and 21 variables were finally selected as the input variables. In this paper, the raw data set is processed by the method of sliding window to obtain W post-sliding window samples, and the first 80% of the post-sliding window samples will be used as historical samples and the last 20% as the testing set. The size of W is finally determined to 2000. Due to the large amount of data, for a better presentation, this paper divides the testing set into G1 and G2, which account for 50% of the testing set respectively.

The configurations of the simulation computer are as follows: the operating system is Windows 11; the CPU is an

TABLE 1. Comparison of modeling accuracy and efficiency with different window size.

Window size	MAE(°C)	RMSE(°C)	Time(s)
19	1.2028	1.7958	37.4568
20	1.0815	1.6730	34.6181
21	1.1423	1.8209	34.2437
22	1.2369	1.9447	36.8024
23	1.1893	1.8737	35.8557
24	1.1994	1.8122	44.5178

TABLE 2. Comparison of modeling accuracy and efficiency with different cluster number.

Cluster number	MAE(°C)	RMSE(°C)	Time(s)
3	1.0214	1.6070	52.3579
4	1.0861	1.6647	34.6358
5	1.0907	1.6953	27.6113
6	1.2637	2.2478	25.2019
7	1.1709	1.8457	23.9330
8	1.2004	1.8631	21.7740

TABLE 3. Comparison of modeling accuracy and efficiency with different number of local modeling samples.

N	MAE(°C)	RMSE(°C)	Time(s)
5	1.2882	2.2068	23.0869
10	1.2216	1.9199	24.1379
15	1.1664	1.9798	22.8319
20	1.1685	1.8422	26.1235
25	1.0493	1.7206	25.9467
30	1.1858	1.9274	26.6824

Intel i5-10440 (2.90 GHz); the RAM is 16 GB; and the code software is Python 3.9.0.

To evaluate the prediction performance of the proposed modeling framework, the root-mean-squared error (RMSE), mean absolute error (MAE), maximum absolute value error (MAX), and the decision coefficient (R^2) are used as the performance indices, which are defined as follows:

$$\begin{cases} \text{RMSE} = \sqrt{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \Big/ N_{test}} \\ \text{MAE} = \sum_{i=1}^{n} \left|\hat{y}_i - y_i\right| \Big/ N_{test} \\ \text{MAX} = \max \left|\hat{y}_i - y_i\right| \qquad , i = 1, 2, \ldots, N_{test} \\ \text{R}^2 = 1 - \text{MSE}(\hat{y}_i, y_i) \Big/ \text{Var}(y_i) \end{cases} \tag{36}$$

where N_{test} is the number of samples in the testing data set; $\hat{y}$ represents the prediction value of the furnace temperature; y represents the actual value of the furnace temperature; Var(y) represents the variance of y, and $\text{MSE}(\hat{y}, y) = \text{RMSE}^2(\hat{y}, y)$.

To perform the DFC-DLWLSTM algorithm, the window size and stride of the sliding window are the first parameters to be determined. In order to fully extract the time series information within each sample, the stride of the sliding window is set to 1. The window size of the sliding window determines the number of points in per sample and is also an important parameter of the LSTM network, which determines the maximum number of historical correlation points of the LSTM. Table 1 presents the modeling accuracy and efficiency of the DFC-DLWLSTM model in the first 100 samples in the testing set G1 when the window size is selected among the candidate length set {19, 20, 21, 22, 23, 24}. The average time for querying samples in modeling and prediction is used as efficiency indices. The window size is determined to be 20.

After determining the window size and stride of the sliding window, the number of clusters of the DFC algorithm will be determined. Table 2 shows the prediction error of the DFC-DLWLSTM model in the first 100 samples in the testing set G1 when the number of clusters is chosen among the candidate length set {3, 4, 5, 6, 7, 8}. As can be seen from Table 2, when the number of clusters increases from 3 to 5, the total time used for modeling and prediction decreases significantly by 47.26%, i.e., by 24.7456 s. Meanwhile, the modeling error indices increase slowly and modestly: MAE increases by 0.0693Â°C (6.78%) and RMSE increases by 0.0883Â°C (5.49%). The final number of clusters is determined to be 5. The structure of the 1D-CNN used in the DFC operating condition classification algorithm is shown in Figure 11.

After determining the number of clusters and the parameters related to the sliding window and dividing the training and testing sets, the DFC algorithm is ready to run. Since the data to be classified do not have real categories, the effectiveness of the DFC algorithm cannot be determined by comparing the predicted values with the real values. The effectiveness of the DFC algorithm can only be determined by analyzing the cluster results of the training set and the deviation of the final furnace temperature prediction between DLWLSTM and DFC-DLWLSTM. Figure 12 shows the time series of each category after the training set is clustered into five categories. To show the shape of the time series more visually, not all time series are shown, but 20 consecutive time series of furnace temperature randomly selected from each of the five training set categories after clustering. As the figure illustrates, the similarity of time series shapes within each category is high, while the similarity of time series between categories is small. The training set for each category represents the trend of the furnace temperature at each of the 20 sampling points: Cluster 1 represents an increasing and then decreasing furnace temperature, while Cluster 2 represents a decreasing and then increasing furnace temperature, both of which mean that the working condition of the furnace is changing. Cluster 3 is a condition that the furnace working with temperature fluctuations; Cluster 4 is a furnace working condition where the furnace temperature is increasing, and Cluster 5 is a furnace working condition where the furnace temperature is decreasing. Then the CNN trained from the clustered and category-labeled training set will have the

TABLE 4. Comparison of modeling accuracy and efficiency with different methods.

Dataset	Method	MAE(°C)	RMSE(°C)	MAX	MAPE	R^2	Time(s)
G1	BP	6.1579	7.9056	17.3592	0.0091	0.9912	\
	RNN	5.0696	7.3638	17.5652	0.0076	0.9925	\
	LSTM	4.5826	7.0254	25.2327	0.0067	0.9932	\
	DLWLSTM	1.0242	1.5972	6.1305	0.0015	0.9996	102.1127
	DFC-DLWLSTM	1.0387	1.6124	7.1547	0.0015	0.9996	30.5112
G2	BP	4.6535	5.9034	17.7719	0.0056	0.9748	\
	RNN	2.6154	3.4717	13.3188	0.0032	0.9913	\
	LSTM	2.5257	3.4040	14.9677	0.0030	0.9916	\
	DLWLSTM	1.8361	2.8531	15.0140	0.0022	0.9941	101.3776
	DFC-DLWLSTM	1.9707	2.8630	14.3684	0.0023	0.9941	30.5385

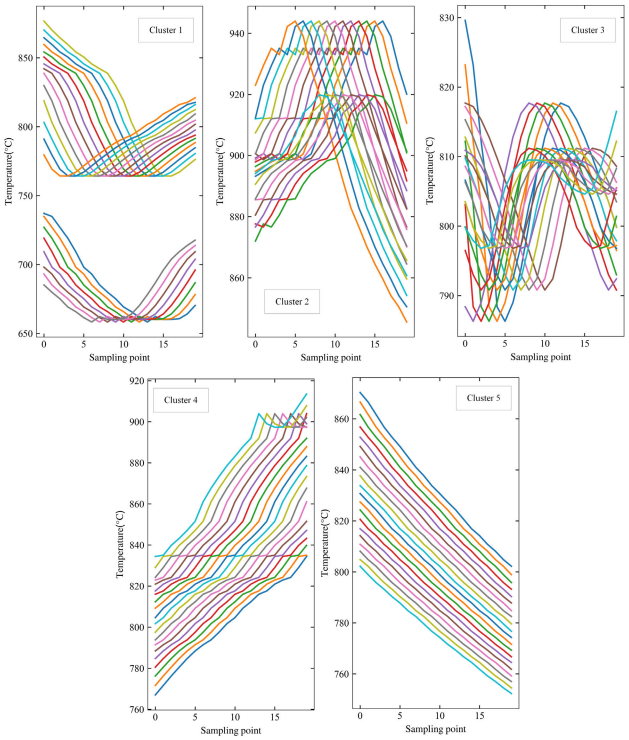


FIGURE 12. 5 categories clustered from the training set.

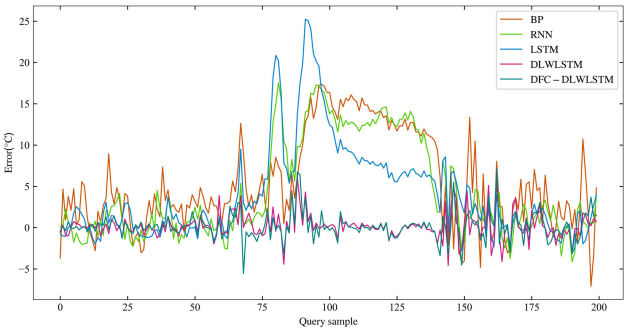


FIGURE 13. The detailed prediction error of different methods.

ability to predict the working condition of the query samples. The final validation of the DFC will be reflected in the comparison of the furnace temperature prediction accuracy of DLWLSTM and DFC-DLWLSTM in the later section.

The number of local modeling samples is an important index in the local modeling process, and different numbers of local modeling samples have a great impact on the accuracy of the local model. Table 3 presents the model accuracy in the first 100 samples in the testing set G1 when the number of local modeling samples N is selected among the candidate length set {5, 10, 15, 20, 25, 30}. The final number of local modeling samples N is set to 25. In this paper, the sample weight adjustment parameter σ is an adaptive parameter whose value is 48 times the maximum similarity value with the query sample among the selected N local modeling samples. The LSTM structure used is shown in Figure 11(b).

For performance comparison, the BP neural network, RNN, LSTM, proposed DLWLSTM, and proposed DFC-DLWLSTM are developed for soft sensors of quality prediction on two different groups of data set G1 and G2, and Table 4 presents the performance of the methods mentioned above on G1, G2. As can be seen from Table 4, the BP neural networks has the worst performance, because although BP neural network has the ability to model nonlinear relationships in process data, this static modeling approach does considers the temporal dynamic information of the data. RNN performs much better than BP neural network because of their ability to acquire dynamic features of data temporal series. But there are limitations for RNN when dealing with long time series. For LSTM, the nonlinear activation function of its units gives it a strong ability to handle nonlinearities, while its selective memory feature gives it the ability to acquire features of long time series; thus, the performance of LSTM is better than that of RNN. However, the prediction results of the offline LSTM are not satisfactory in the face of the time-varying aluminum smelting process. The proposed DLWLSTM method uses the DTW distance as a measure of the similarity between query samples and historical samples, builds a corresponding soft sensor model for each query sample, and uses the similarity between local modeling samples and query samples as the weight of the local modeling samples. The aforementioned strategy adopted by DLWLSTM effectively enhances the ability of the soft sensor to deal with the time-varying process of

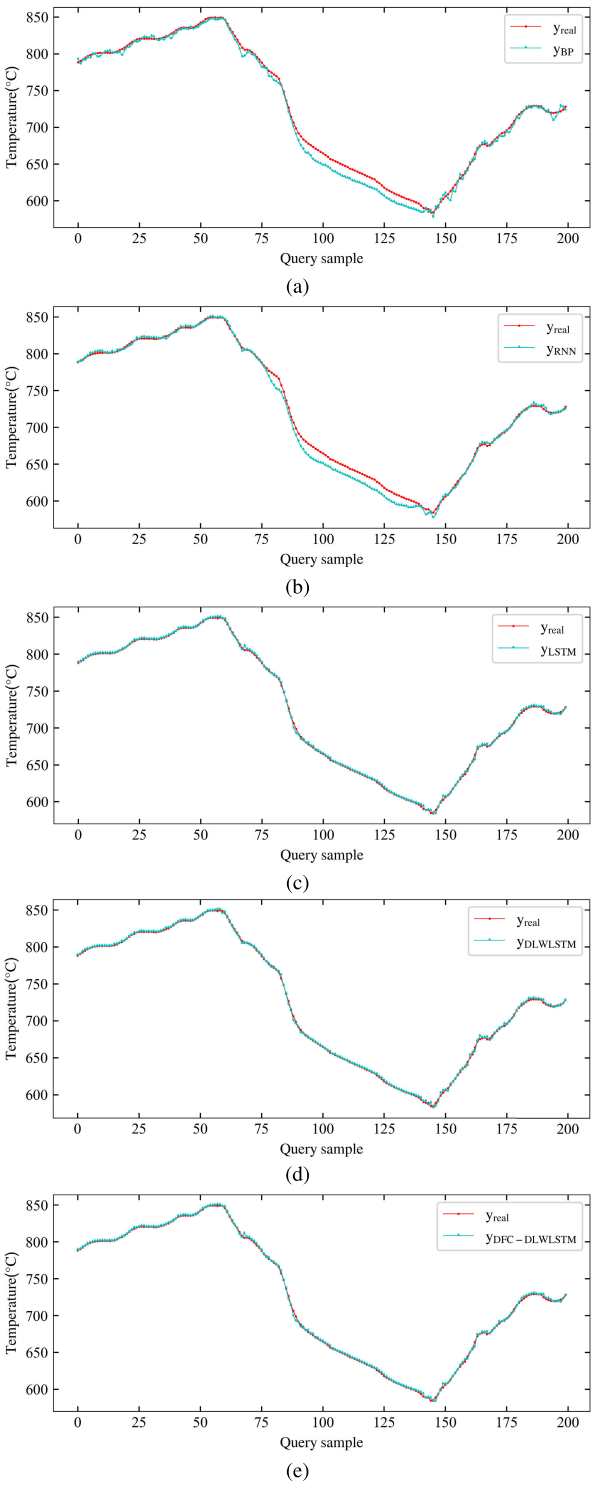


FIGURE 14. The detailed prediction results of different methods on G1: (a) BP; (b) RNN; (c) LSTM; (d) DLWLSTM; (e) DFC-DLWLSTM.

aluminum smelting. The proposed DFC-DLWLSTM adopts the strategy of classifying first and then modeling, so that each query sample has a corresponding category of historical sample set, which effectively reduces the number of samples in the historical sample set of each query sample, thereby greatly reducing the time consumption in the query sample's modeling and prediction.

Figure 13 shows the detailed prediction errors of the five methods mentioned above. To better demonstrate the performance of the five methods, the detailed prediction results are shown in Figure 14, and the dataset used is the testing set G1. It can be seen in Figure 14 that the prediction curve of DFC-DLWLSTM are able to follow the curve of the true value better than those of the BP neural network, RNN and LSTM. From a comprehensive comparison of the DLWLSTM model and DFC-DLWLSTM in Table 4 and Figure 14, we can see that the DFC-DLWLSTM model can significantly increase the modeling efficiency while retaining the prediction accuracy.

V. CONCLUSION

To deal with the problems in predicting the furnace temperature of a regenerative aluminum smelting furnace, a soft sensor modeling method based on DFC-DLWLSTM is proposed. This modeling method fully extracts the temporal characteristics of the data by employing LSTM neural networks. By the method of locally sample-weighted modeling based on DTW distance, the local model for the query samples is built while considering the weights of different historical input samples, which not only solves the time-varying problem, but also effectively extracts the nonlinear features of the input samples. In addition, the historical sample categories are classified by the DFC operating condition clustering algorithm, which reduces the time used in local modeling while the prediction accuracy can be maintained. The effectiveness of the proposed method is verified by the aluminum smelting process data from an industrial plant.

REFERENCES

[1] M. I. Hassan and R. Al Kindi, "Feasibility study of regenerative burners in aluminum holding furnaces," *JOM*, vol. 66, no. 9, pp. 1603–1611, Sep. 2014.

[2] Y. A. Shardt, *Statistics for Chemical and Process Engineers*. Cham, Switzerland: Springer, 2015.

[3] Z.-N. Li, M.-S. Chu, Z.-G. Liu, G.-J. Ruan, and B.-F. Li, "Furnace heat prediction and control model and its application to large blast furnace," *High Temp. Mater. Processes*, vol. 38, no. 2019, pp. 884–891, Feb. 2019.

[4] A. O. Nieckele, M. F. Naccache, and M. S. P. Gomes, "Combustion performance of an aluminum melting furnace operating with natural gas and liquid fuel," *Appl. Thermal Eng.*, vol. 31, no. 5, pp. 841–851, Apr. 2011.

[5] J.-M. Wang, P. Xu, H.-J. Yan, J.-M. Zhou, S.-X. Li, G.-C. Gui, and W.-K. Li, "Burner effects on melting process of regenerative aluminum melting furnace," *Trans. Nonferrous Met. Soc. China*, vol. 23, no. 10, pp. 3125–3136, Oct. 2013.

[6] L. Qiu, Y. Feng, Z. Chen, Y. Li, and X. Zhang, "Numerical simulation and optimization of the melting process for the regenerative aluminum melting furnace," *Appl. Thermal Eng.*, vol. 145, pp. 315–327, Dec. 2018.

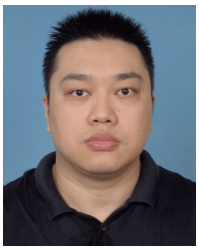
[7] J. Dai, N. Chen, X. Yuan, W. Gui, and L. Luo, "Temperature prediction for roller kiln based on hybrid first-principle model and data-driven MW-DLWKPCR model," *ISA Trans.*, vol. 98, pp. 403–417, Mar. 2020.

[8] M. Kano, K. Miyazaki, S. Hasebe, and I. Hashimoto, "Inferential control system of distillation compositions using dynamic partial least squares regression," *J. Process Control*, vol. 10, nos. 2–3, pp. 157–166, Apr. 2000.

[9] B. Liang, X. Yuan, and Z. Ge, "Co-training partial least squares model for semi-supervised soft sensor development," *Chemometrics Intell. Lab. Syst.*, vol. 147, pp. 75–85, Oct. 2015.

[10] J. Zheng and Z. Song, "Mixture modeling for industrial soft sensor application based on semi-supervised probabilistic PLS," *J. Process Control*, vol. 84, pp. 46–55, Dec. 2019.
[11] N. Chen, J. Dai, X. Yuan, W. Gui, W. Ren, and H. N. Koivo, "Temperature prediction model for roller kiln by ALD-based double locally weighted kernel principal component regression," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 8, pp. 2001–2010, Aug. 2018.
[12] X. Yuan, Z. Ge, B. Huang, Z. Song, and Y. Wang, "Semisupervised JITL framework for nonlinear industrial soft sensing based on locally semisupervised weighted PCR," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 532–541, Apr. 2017.
[13] L. Wang, C. Yang, Y. Sun, H. Zhang, and M. Li, "Effective variable selection and moving window HMM-based approach for iron-making process monitoring," *J. Process Control*, vol. 68, pp. 86–95, Aug. 2018.
[14] X. Chen, J. Dai, and Y. Luo, "Temperature prediction model for a regenerative aluminum smelting furnace by a just-in-time learning-based triple-weighted regularized extreme learning machine," *Processes*, vol. 10, no. 10, p. 1972, Sep. 2022.
[15] W. Shao, Z. Ge, and Z. Song, "Bayesian just-in-time learning and its application to industrial soft sensing," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2787–2798, Apr. 2020.
[16] L. Zhang, S. Wang, X. Zhang, Z. He, J. Huang, and L. Qi, "Temperature prediction model for rotary kiln based JITL with regularized extreme learning machine," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 5806–5810.
[17] H. Wei, Y. Xu, and R. Zhang, "Neural networks based model predictive control of an industrial polypropylene process," in *Proc. Int. Conf. Control Appl.*, vol. 1, 2002, pp. 397–402.
[18] J. C. B. Gonzaga, L. A. C. 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, 2009.
[19] X. Li, F. Zhang, G. Wang, and F. Fang, "Joint optimization of statistical and deep representation features for bearing fault diagnosis based on random subspace with coupled LASSO," *Meas. Sci. Technol.*, vol. 32, no. 2, Feb. 2021, Art. no. 025115.
[20] F. Curreri, L. Patanè, and M. G. Xibilia, "RNN- and LSTM-based soft sensors transferability for an industrial process," *Sensors*, vol. 21, no. 3, p. 823, Jan. 2021.
[21] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[23] Z. Li, W. Jiang, S. Zhang, D. Xue, and S. Zhang, "Research on prediction method of hydraulic pump remaining useful life based on KPCA and JITL," *Appl. Sci.*, vol. 11, no. 20, p. 9389, Oct. 2021.
[24] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. KDD Workshop*, vol. 10, no. 16. Seattle, WA, USA, 1994, pp. 359–370.
[25] M. Müller, "Dynamic time warping," in *Information Retrieval for Music and Motion*. Berlin, Germany: Springer, 2007, pp. 69–84.
[26] H. Izakian, W. Pedrycz, and I. Jamal, "Fuzzy clustering of time series data using dynamic time warping distance," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 235–244, Mar. 2015.

YANHUI DUAN received the B.S. degree in electrical engineering and automation from the Guangdong University of Technology, in 2020. He is currently pursuing the M.S. degree in electrical engineering with the School of Electrical Engineering, Guangxi University. His research interests include soft sensors, machine learning, and the optimization of complex systems.



JIAYANG DAI received the B.S., M.S., and Ph.D. degrees from Central South University, in 2009, 2012, and 2019, respectively. He is currently an Assistant Professor with the School of Electrical Engineering, Guangxi University. His research interests include soft sensor, machine learning, reinforcement learning and optimization, and the control of complex systems.

YASONG LUO received the B.S. degree in electrical engineering and automation from the Chengdu University of Information Technology, in 2019. He is currently pursuing the M.S. degree in energy power (electrical engineering) with the School of Electrical Engineering, Guangxi University. His research interests include soft sensors, intelligent algorithms, modeling, and the optimization of complex systems.

GUANYUAN CHEN received the B.S. degree in electrical engineering and automation from the Shanghai University of Electric Power, in 2020. He is currently pursuing the M.S. degree in electrical engineering with the School of Electrical Engineering, Guangxi University. His research interests include MMC and machine learning.

XINCHEN CAI received the B.S. degree in electronic and information engineering from the Hubei University of Arts and Science, in 2019. He is currently pursuing the M.S. degree in electrical engineering with the School of Electrical Engineering, Guangxi University. His current research interests include computer vision, deep learning applications, and SoC design for deep neural networks.