Deep Learning With Spatiotemporal Attention-Based LSTM for Industrial Soft Sensor Model Development

Xiaofeng Yuan, *Member, IEEE*, Lin Li, Yuri A. W. Shardt, Yalin Wang, *Member, IEEE*, and Chunhua Yang, *Member, IEEE*

***Abstract*—Industrial process data are naturally complex time series with high nonlinearities and dynamics. To model nonlinear dynamic processes, a long short-term memory (LSTM) network is very suitable for soft sensor model development. However, the original LSTM does not consider variable and sample relevance for quality prediction. In order to overcome this problem, a spatiotemporal attention-based LSTM network is proposed for soft sensor modeling, which can, not only identify important input variables that are related to the quality variable at each time step, but also adaptively discover quality-related hidden states across all time steps. By taking the spatiotemporal quality-relevant interactions into consideration, the prediction performance can be improved for the soft sensor model. The effectiveness and flexibility of the proposed model is demonstrated on an industrial hydrocracking process to predict the initial boiling points of heavy naphtha and aviation kerosene.**

***Index Terms*—Attention mechanism, deep learning, quality prediction, soft sensor, spatiotemporal attention-based long short-term memory (LSTM) (STA-LSTM).**

I. Introduction

IN INDUSTRIAL processes, product properties are important to ensure process safety and production quality [1]–[4]. In many situations, offline laboratory analysis is used for some process quality variables that are difficult to obtain in real time. Although offline laboratory analysis can provide accurate measurements, the sampling cycle is often very long, and it usually leads to low sampling rates and large measurement delay. Some plants may be installed with online analytical instruments to obtain measurement data for key quality variables. However, online quality instruments are often expensive and difficult to maintain. Neither offline tests nor online sensors can meet the requirements for real-time process monitoring, control, and optimization. Recently, soft sensors have been widely used for online estimation of key product qualities thanks to their rapid response, low maintenance costs, and accurate prediction results. Soft sensors can predict the difficult-to-measure quality variables by building predictive mathematical models based on secondary process variables that are easy to measure, such as temperatures, pressures, and flowrate [5]–[8].

Usually, soft sensors are divided into two main categories: first principle, or white-box models, and data-driven, or black-box models. First-principle models are developed using the laws of nature, such as mass and energy balances, force balances, or reaction mechanisms [9]. On the other hand, data-driven models are developed solely using the available data without necessarily considering the physical meaning of the resulting models [9]. Given the common use of distributed control systems and the relative availability of historical process data, data-driven soft sensors have become popular. Typical data-driven modeling methods are principal component analysis [10], [11] partial least squares [12], [13] and artificial neural network (ANN) [14], [15]. Of these, ANN is a common method that is widely used in soft sensor development. For example, Dam *et al.* [16] proposed a soft sensor based on ANNs and introduced a genetic algorithm to optimize the network structure and weights. Napoli *et al.* [17] developed a soft sensor for predicting the atmospheric pylon aviation kerosene condensation point using self-introduction sampling, noise injection, and neural network stacking. However, shallow ANNs have limited ability to express complex functions, and its generalizability is restricted to large systems. Multilayer networks are easily affected by gradient vanishing and exploding problems. In 2006, Hinton *et al.* [18] proposed a deep-learning technique to resolve this problem. They showed that an ANN with multiple hidden layers can accurately extract features using deep learning. Also, it is possible for deep neural networks to effectively overcome the difficulty of network training through layer-wise unsupervised pretraining and supervised fine tuning.

Deep neural networks have also been introduced for soft sensor modeling because of their better performance [19]–[27]. However, these soft sensors are mainly based on static deep

Manuscript received November 7, 2019; revised February 25, 2020; accepted March 16, 2020. Date of publication April 9, 2020; date of current version January 27, 2021. This work was supported in part by the Program of National Natural Science Foundation of China under Grant 61590921, 61860206014, U1911401, 61703440, in part by the National Key R&D Program of China under Grant 2018YFB1701100, 2018AAA0101603, and in part by the Natural Science Foundation of Hunan Province of China under Grant 2018JJ3687. *(Corresponding authors: Lin Li; Yalin Wang.)*

Xiaofeng Yuan, Lin Li, Yalin Wang, and Chunhua Yang are with the School of Automation, Central South University, Changsha 410083, China (e-mail: yuanxf@csu.edu.cn; 1192732314@qq.com; ylwang@csu.edu.cn; ychh@csu.edu.cn).

Yuri A. W. Shardt is with the Department of Automation Engineering, Technical University of Ilmenau, Ilmenau 98684, Germany (e-mail: yuri.shardt@tu-ilmenau.de).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2020.2984443

networks, such as deep belief networks (DBN) [28] and stacked autoencoders (SAE) [29], [30], in which data samples are assumed to be independent and identically distributed (the *i.i.d.* assumption). Nevertheless, industrial processes have intrinsically complex nonlinear dynamic behavior due to such factors as complex physiochemical reaction mechanisms, feedback control, and dynamic noise. The data sequences are sampled from real-valued and continuous processes over time. Hence, they are naturally time sequences with highly nonlinear temporal correlations. To model such data sequences, the models have to include more data from the past steps or have memory unit of the past inputs. Recurrent neural networks (RNNs) [31], a type of dynamic neural networks, have been widely used to capture temporal dynamic behavior in time series data. However, it is difficult for standard RNNs to model long sequences, since they also suffer from the gradient vanishing problem [32]. To handle this issue, a long short-term memory (LSTM) network was proposed by Hochreiter and Schmidhuber [33], in which memory cells and three nonlinear gates are used to replace the basic activation unit in RNN. LSTM can not only forget the useless information in the past, but also judge the current information and store useful information in the memory cell [34], [35]. Thus, LSTM networks are more effective in learning long-term temporal dependencies. This leads to its successful applications in many fields such as language modeling [36], time series prediction [37], and automatic speech recognition [38].

Although LSTM is very helpful in capturing long-term dependencies, it cannot focus on different variables at different time steps. To overcome this problem, an attention-based encoder–decoder network was proposed in [39]. Based on LSTM units, an encoder–decoder network can be constructed to resolve the sequence-to-sequence problem. With the encoder part, the input sequence can be converted into a fixed-length vector, and then the generated fixed vector is converted into an output sequence by the decoder. However, the performance of this model will decrease rapidly with the increase of the length of the input sequence. This is a common problem in industrial data sequences, since the quality variables are often predicted based on very long lagged input series. The attention-based encoder–decoder architecture seeks to differentiate between hidden states with different attention weights across all time steps in a prediction window. In recent years, attention mechanisms have performed well for many different tasks like machine translation, image classification, and natural language processing [40], [41]. The attention phenomenon is also very common in industrial processes, since data samples at previous instants always have a different impact on prediction of the current data. However, this is rarely considered in the previous process data modeling approaches. In this article, the attention mechanism is used for dynamic modeling of the industrial processes. Moreover, most existing attention-based models mainly consider the temporal dynamics and correlations between data samples. The impact of the input variables on the quality prediction is not considered in the existing attention mechanism. For industrial processes, the secondary process variables often have different impact on the quality prediction due to the physiochemical properties of the system.

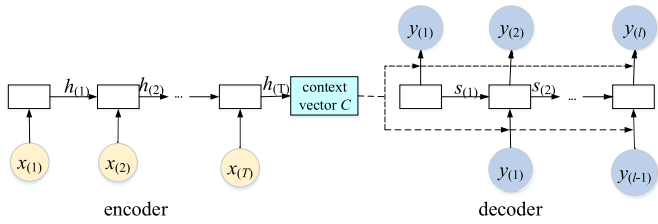

Fig. 1. Structure of an encoder–decoder.

Thus, this article proposes a new spatiotemporal, attention-based LSTM (STA-LSTM) method for application to industrial soft sensor modeling. In this approach, the attention mechanism seeks to obtain the spatial correlation between the input and target quality variables. Then, the temporal attention mechanism seeks to model the time dynamic behavior for the final prediction. This allows STA-LSTM to not only adaptively identify the input features, but also handle dynamic behavior. Finally, the proposed STA-LSTM method will be applied to an industrial hydrocracking process to predict the initial boiling points of heavy naphtha and aviation kerosene.

II. Encoder–Decoder Architecture and the Attention Mechanism

An encoder–decoder is a common framework in deep learning. It was proposed by Sutskever [42] to solve the problem of sequence-to-sequence modeling. For the encoder, the input sequence is first transformed to the hidden state sequence. Then, the hidden state sequence is converted into a fixed-length vector. After that, the previously generated fixed vector is used to predict a target output sequence by the decoder. The structure of the encoder–decoder architecture is shown in Fig. 1. The basic encoder and decoder units can be any models, such as SAE, RNN, or LSTM. Moreover, the encoder and decoder units can be different from each other.

The encoder–decoder network is often carried out in a sliding window manner. Assume the input and output sequences are, respectively, $\{x_{(1)}, x_{(2)}, \ldots, x_{(T)}\}$ and $\{y_{(1)}, y_{(2)}, \ldots, y_{(l)}\}$ in a given window. For soft sensor application, we often have $T = l$. However, they can be different for other modeling tasks like machine translation, in which the length of the output sequences is not known *a priori*.

For the encoder–decoder architecture, the input sequences are first encoded to the hidden series as $\{h_{(1)}, h_{(2)}, \ldots, h_{(T)}\}$. The feature representation $h_{(t)}$is progressively learned from $x_{(t)}$ and $h_{(t-1)}$ as

$$h_{(t)} = f(x_{(t)}, h_{(t-1)}) \tag{1}$$

where $f(\cdot)$ could be any nonlinear activation function like the sigmoid or tan*h* function. As can be seen, the hidden state series $\{h_{(1)}, h_{(2)}, \ldots, h_{(T)}\}$are dynamic features extracted from the input sequence. Then, they may be directly used to serve as

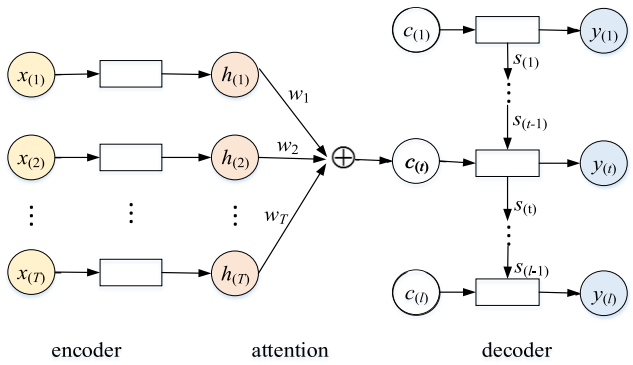

Fig. 2. Structure of an attention-based encoder–decoder.

the inputs for the decoder to estimate the output sequence of $y_{(t)}$ with $h_{(t)}$ if the lengths are the same for the input and output sequences. However, it is difficult to directly apply this method when the input and output sequences have different lengths with complicated and nonmonotonic relationships. In this way, a simple strategy is to further transfer the hidden state sequence into a fixed-sized context vector. In this way, the context vector c can be generated from the hidden state sequence $\{h_{(1)}, h_{(2)}, \ldots, h_{(T)}\}$ as

$$c = f_1(h_{(1)}, h_{(2)}, \ldots, h_{(T)}) \tag{2}$$

where $f_1(\cdot)$ represents a mapping function. In this way, the context vector includes the information across the whole input sequence, which can be regarded as a deeper abstract of the input sequence. Then, the content vector can be used as the input for the decoder to predict the output sequence. Once the context vector is obtained, the output at time t can be progressively predicted by its previous output sequence $\{y_{(1)}, y_{(2)}, \ldots, y_{(t-1)}\}$ and the vector c as

$$\hat{y}_{(t)} = f_2(c, y_{(1)}, \ldots, y_{(t-1)}) \;\; 1 \le t \le l \tag{3}$$

where $f_2(\cdot)$ is also a nonlinear activation function.

However, with the increase in the length of the input sequence, the context vector will lose long-term input information about the past, which leads to a decrease in the prediction performance [43]. To avoid this problem, the attention mechanism is introduced to the encoder–decoder framework. The structure of the attention-based encoder–decoder model is shown in Fig. 2. For the attention mechanism, a distinct vector $c_{(t)}$ rather than a fixed c is designed to predict each output sample $y_{(t)}$ at all instants. For prediction at sampling step t, the attention mechanism assigns an individual attention weight w_i to each encoded hidden feature state $h_{(i)}$ according to its relationship with the previous hidden state in the decoder. Thus, the context vector changes for different predicted output instants. The context vector at time t can be computed as

$$c_{(t)} = \sum_{i=1}^{T} w_i h_{(i)} \tag{4}$$

where w_i is the attention weight of the ith hidden state $h_{(i)}$ at time i.

III. Spatiotemporal Attention-Based LSTM (STA-LSTM) Network

In this section, the spatiotemporal attention-based LSTM network is developed for soft sensor modeling. Following the encoder–decoder architecture, two kinds of attention mechanisms are used in the proposed STA-LSTM model. In the encoder, a spatial variable attention mechanism is introduced to selectively distinguish input variables related to quality prediction at each time step. Moreover, different spatial attention values are assigned to the input variables. Then, the variable attention-weighted sample data becomes the new input to the encoder LSTM. After that, the encoder LSTM network is used to learn the hidden states of the new weighted inputs. At the second step, a temporal attention mechanism is introduced to adaptively find out the encoder hidden states related to the quality prediction across different time steps. The adaptive context vector is the weighted sum over the products of the encoder hidden states and their corresponding temporal attention values. Finally, with the adaptive context vector as the input, the output is predicted by another decoder LSTM. Fig. 3 illustrates the proposed model. Detailed steps are described below.

A. Spatial Attention

Since it is necessary to capture the long-term dependencies in the industrial data-time series, LSTM units are used as the basic activation function units in the proposed STA-LSTM model. For the given input sequence $\{x_{(1)}, x_{(2)}, \ldots, x_{(T)}\}$ in a subwindow, assume each sample has n input variables with $x_{(t)} = (x^1_{(t)}, x^2_{(t)}, \ldots, x^n_{(t)})$. We can obtain the relationship between each input variable and the quality variable by referring to the previous decoder hidden state, which can be calculated by a metric similarity between the current original input $x_{(t)}$ and the previous hidden state $s_{(t-1)}$in the decoder LSTM. The spatial attention architecture can be a multilayer perceptron network. Usually, a two-layer network is used to obtain the variable spatial attention as

$$e^i_{(t)} = V^i_1 \tanh(W^i_1 s_{(t-1)} + U^i_1 x^i_{(t)} + b^i_1) \;\; 1 \le i \le n \tag{5}$$

$$\alpha^i_{(t)} = \frac{\left|e^i_{(t)}\right|}{\sum_{j=1}^{n}\left|e^j_{(t)}\right|}, \qquad 1 \le i \le n \tag{6}$$

where V^i_1, W^i_1, U^i_1, and b^i_1 are the parameters to be learnt, tan$h$ is the hyperbolic activation function, and $e^i_{(t)}$ is the attention value that represents the importance of the ith input variable at time t for quality prediction. Equation (6) is mainly used to ensure that the attention values for all the input variables add up to 1 at each sampling instant t. The normalized value $\alpha^i_{(t)}$ is called the spatial attention weight. $|\bullet|$ represents taking the absolute value. Since $s_{(t-1)}$ is the hidden state of the decoder LSTM, it contains information related to the quality variable. As can be seen from Fig. 3, the spatial attention architecture is usually a perceptron network, the parameters in (5) are learnt through back propagation through time (BPTT) in the training procedure of the whole network. Once the parameters of the spatial attention network are determined, the correlation between input and

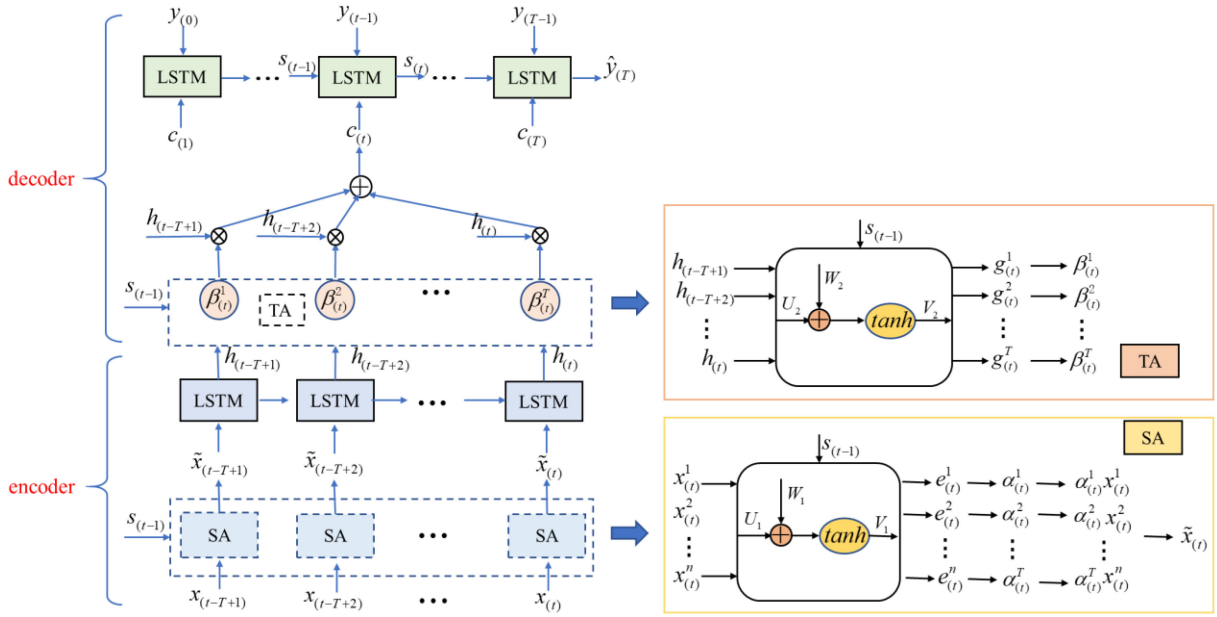

Fig. 3. Framework of the spatiotemporal attention-based LSTM (TA: temporal attention; SA: spatial attention).

output can be obtained by computing a similarity calculation between the decoder hidden state $s_{(t-1)}$ and input data $x_{(t)}$ with (5). Correspondingly, the attention weights can be directly calculated with (5) and (6).

When the spatial attentions are obtained for the input variables of sample $x_{(t)}$, the spatial attention weighted sample can be expressed as

$$\tilde{x}_{(t)} = (\alpha^1_{(t)} x^1_{(t)}, \alpha^2_{(t)} x^2_{(t)}, \ldots, \alpha^n_{(t)} x^n_{(t)}). \tag{7}$$

A LSTM network is then used to learn the hidden states of the encoder from the variable attention weighted input $\tilde{x}_{(t)}$. In LSTM, three gate controllers and a memory cell are placed into the basic LSTM unit, namely the input, forget, and output gates. The three gates are used to determine what information should be remembered from the weighted time series. The memory cell is used to store the input information for all time steps. The LSTM network implements temporal memory through the switch of these gates and a memory cell to prevent the gradient vanishing problem. Then, the hidden state can be written during a forward pass as

$$f_{(t)} = \sigma(W_{fx}\tilde{x}_{(t)} + W_{fh}h_{(t-1)} + b_f) \tag{8}$$

$$i_{(t)} = \sigma(W_{ix}\tilde{x}_{(t)} + W_{ih}h_{(t-1)} + b_i) \tag{9}$$

$$o_{(t)} = \sigma(W_{ox}\tilde{x}_{(t)} + W_{oh}h_{(t-1)} + b_o) \tag{10}$$

$$\tilde{c}_{(t)} = \tanh(W_{cx}\tilde{x}_{(t)} + W_{ch}h_{(t-1)} + b_c) \tag{11}$$

$$m_{(t)} = f_{(t)} \odot m_{(t-1)} + i_{(t)} \odot \tilde{c}_{(t)} \tag{12}$$

$$h_{(t)} = o_{(t)} \odot \tanh(m_{(t)}) \tag{13}$$

where $f_{(t)}$, $i_{(t)}$, $o_{(t)}$, $\tilde{c}_{(t)}$ represent the forget gate, input gate, output gate, and intermediate state of the encoder LSTM unit; $m_{(t)}$ is the corresponding memory cell; $\odot$ is pointwise multiplication of two vectors; $\sigma$is the nonlinear sigmoid activation functions; and W_{f*}, W_{o*}, W_{i*}, W_{c*} and b_f, b_i, b_o, b_c are the parameters to be learnt. By introducing the spatial variable attention mechanism, the encoder can adaptively identify the input variables that are more related to the quality variable. After the spatial-attention-based LSTM, the hidden states are used as inputs to the temporal-attention-based LSTM.

B. Temporal Attention

After the encoder, another LSTM neural network-based decoder is used to predict the quality variable $\hat{y}_{(t)}$. With the increase of the input sequence, it is difficult to retain all the necessary information for the decoder. Thus, the performance of the encoder–decoder architecture will degrade rapidly as the input sequence increases. To solve this problem, the temporal sample attention mechanism is introduced to the decoder LSTM to adaptively determine relevant hidden states generated from the encoder LSTM across all time instants, which can be measured by referring to the previous decoder hidden state. Each encoder hidden state is assigned a temporal attention value. Then, an adaptively weighted content vector is obtained as the input for the decoder LSTM. In this way, the attention mechanism breaks the limitation of the traditional encoder–decoder structure that internally relies on a fixed-length vector during encoding and decoding. As can be seen from Fig. 3, the temporal attention value of hidden state at time t can be computed as

$$g^k_{(t)} = V^k_2 \tanh(W^k_2 s_{(t-1)} + U^k_2 h_{(t-T+k)} + b^k_2) \quad 1 \le k \le T \tag{14}$$

$$\beta^k_{(t)} = \frac{\left|g^k_{(t)}\right|}{\sum_{m=1}^{T}\left|g^m_{(t)}\right|}, \quad 1 \le k \le T \tag{15}$$

where $s_{(t-1)}$ is the previous decoder hidden state; V_2^k, W_2^k, U_2^k, and b_2^k are the parameters to be learnt with regard to the kth sample in the window; $g_{(t)}^k$ represents the attention value of the kth encoder hidden state of the subwindow for time step t; and T is the subwindow size. Equation (15) is mainly used to ensure that the attention values for all the hidden states add up to 1. The normalized value $\beta_{(t)}^k$ is called the temporal attention weight. Then, a temporal weighted sum of all the encoder hidden state can be calculated to get the adaptive context vector

$$c_{(t)} = \sum_{k=1}^{T} h_{(t-T+k)} \beta_{(t)}^k. \tag{16}$$

Once we obtain the context vector at time t, it is combined with the given target series $\{y_{(1)}, y_{(2)}, \ldots, y_{(t-1)}\}$ to update the decoder hidden state

$$\tilde{s}_{(t-1)} = W_3 y_{(t-1)} + V_3 s_{(t-1)} + b_3 \tag{17}$$

$$s_{(t)} = f_l(c_{(t)}, \tilde{s}_{(t-1)}) \tag{18}$$

where W_3 and V_3 are weight matrices; b_3 is the bias vector; and $f_l(\cdot)$ is an LSTM unit. Finally, the prediction output $\hat{y}_{(T)}$ is computed using

$$\begin{aligned} \hat{y}_{(T)} &= F(y_{(1)}, y_{(2)}, \ldots, y_{(T-1)}, x_{(1)}, x_{(2)}, \ldots, x_{(T)}) \\ &= V\left(f_l\left(\left[\tilde{s}_{(T-1)}; c_{(T)}\right]\right)\right) + b_v \end{aligned} \tag{19}$$

where $[\tilde{s}_{(T-1)}; c_{(T)}]$ is a concatenation of the decoder hidden state at time $T-1$ and the context vector $c_{(T)}$ at time T; and V and b_v are weight matrix and bias vectors.

For model training, the Adam algorithm is used because it is superior to the momentum gradient descent method and the root mean square back propagation (RMSProp) algorithm [44] in terms of computing time and memory requirements. The parameters of the model can be learnt by minimizing the mean squared error (MSE) using standard BPTT. MSE can be calculated as

$$\text{MSE} = \frac{1}{T_{\text{training}}} \sum_{t=1}^{T_{\text{training}}} \left(y_{(t)} - \hat{y}_{(t)}\right)^2 \tag{20}$$

where $y_{(t)}$ and $\hat{y}_{(t)}$ are the actual and predicted quality values at sampling time t, and T_{training} is the total number of training data.

Algorithm 1 gives the details of the proposed STA-LSTM method. BPTT is used to compute the gradient of parameters. Then, the Adam algorithm is used to update the network parameters.

IV. Case Studies

To validate the effectiveness of STA-LSTM for soft sensor prediction, it is applied to an industrial hydrocracking process to predict the initial boiling points of the heavy naphtha and aviation kerosene. Fig. 4 shows the STA-LSTM-based soft sensor modeling framework. The configurations of the simulation computer are: the operating system is Windows 7; the CPU is an Intel i5-4460 (3.20 GHz); the RAM is 8 GB; and the code software is Python 3.5.

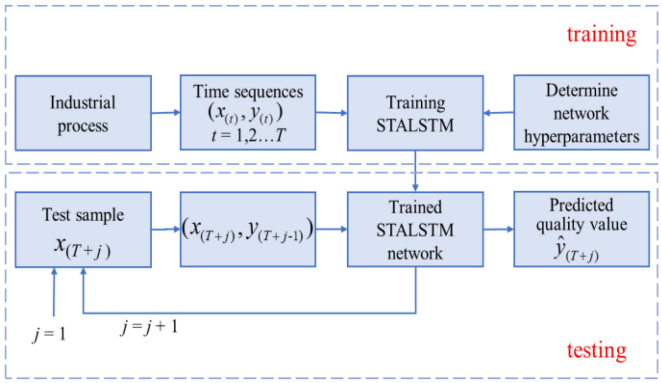

Fig. 4. Flowchart for the STA-LSTM-based soft sensor model framework.

Algorithm 1: Spatiotemporal Attention-based LSTM

Input: Dataset $D = (x_{(t)}, y_{(t)}), t = 1, 2, ..., k$ with $x_{(t)} = \left\{x_{(t)}^1, x_{(t)}^2, ..., x_{(t)}^n\right\}$; Learning rate η; Hidden neurons N; Batch size B; Training epochs E; Window size T

Output: Predicted quality variable $\hat{y}_{(t)}, t = 1, 2, ..., k$

Steps:

Standardized Dataset D;

Random Initialization of Network Parameters;

for $t \leftarrow 1$ **to** k **do**

  Encoder:

  for $m \leftarrow 1$ **to** T **do**

    calculate spatial variable attention $e_{(t-m+1)} \leftarrow V_1 tanh(W_1 s_{(t-m)} + U_1 x_{(t-m+1)})$;

    attention normalization $\alpha_{(t-m+1)}^i \leftarrow \frac{\|e_{(t)}^i\|}{\sum_{j=1}^n \|e_{(t)}^j\|}, i = 1, 2, ..., n$;

    spatial attention weighted sample $\tilde{x}_{(t-m+1)} \leftarrow \alpha_{(t-m+1)} \cdot x_{(t-m+1)}$;

    encoder hidden states $h_{(t-m+1)} \leftarrow f_{LSTM1}(\tilde{x}_{(t-m+1)}, h_{(t-m)})$;

  Decoder:

  for $r \leftarrow 1$ **to** T **do**

    calculate temporal attention $g_{(t)}^r \leftarrow V_2 tanh(W_2 s_{(t-1)} + U_2 h_{(t-r+1)})$;

    attention normalization $\beta_{(t)}^r \leftarrow \frac{\|g_{(t)}^r\|}{\sum_{m=1}^T \|g_{(t)}^m\|}$;

  context vector $c_{(t)} \leftarrow \beta_{(t)}^1 h_{(t)} + \beta_{(t)}^2 h_{(t-1)} + ... + \beta_{(t)}^T h_{(t-T+1)}$;

  decoder hidden states $s_{(t)} \leftarrow f_{LSTM2}(c_{(t)}, s_{(t-1)}, y_{(t-1)})$;

  predicted quality variable $\hat{y}_{(t)} \leftarrow f_{linear}(s_{(t)})$;

For performance comparison, the autoregressive integrated moving average model with external input (ARIMAX), static DBN, RNN, the standard LSTM, the attention-based LSTM without spatial variable attention, and the proposed STA-LSTM are developed for soft sensors of quality prediction. The root mean squared error (RMSE) and correlation coefficient, R^2, are used to compare the accuracy of different models. These are defined as

$$\text{RMSE} = \sqrt{\sum_{t=1}^{T_{\text{testing}}} \left(y_{(t)} - \hat{y}_{(t)}\right)^2 / (T_{\text{testing}} - 1)} \tag{21}$$

$$R^2 = 1 - \sum_{t=1}^{T_{\text{testing}}} \left(y_{(t)} - \hat{y}_{(t)}\right)^2 \Bigg/ \sum_{t=1}^{T_{\text{testing}}} \left(y_{(t)} - \tilde{y}\right)^2 \tag{22}$$

where T_{testing} is the number of testing samples; $\tilde{y}$ is the mean of the real quality values in the testing data; and $y_{(t)}$ and $\hat{y}_{(t)}$

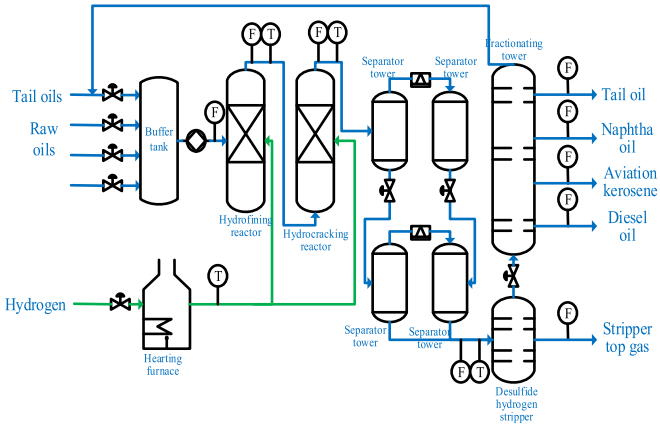

Fig. 5. Schematic of the hydrocracking process.

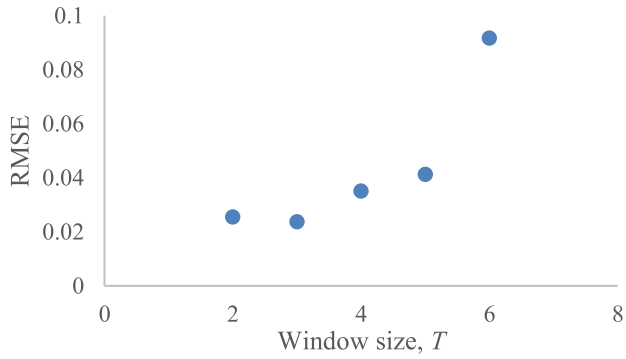

Fig. 6. Relationship between RMSE and window size, *T*, for STA-LSTM for the initial boiling point of heavy naphtha dataset.

are, respectively, the labeled and predicted values for the quality variable at *t*.

A. Description of the Hydrocracking Process

Hydrocracking is an important process in the petrochemical industry that converts the raw oil into its valuable constituents. Fig. 5 shows a diagram of a typical hydrocracking process. As can be seen in Fig. 5, the main devices in this process are the heating furnace, the hydrotreater, the hydrocracker, the high/low-pressure separators (HPS/LPS) and the distillation columns in the downstream separation part. The heavy gas oil and vacuum gas oil are hydrogenated, cracked, and isomerized with a hydrogen-rich gas stream at high pressure and temperature. In this way, the heavy compounds are converted into light oil products such as gasoline, kerosene, and light diesel [45]. Concretely, the make-up hydrogen and heavy vacuum gas oil are first mixed as the two main raw materials to carry out hydrocracking reaction under high temperature and pressure in the hydrocracking reactor. Then, the light oil products such as gasoline, kerosene, and light diesel oil can be obtained through a series of heat transfer, cooling, and heating in the HPS/LPS and fractionation unit. As can be seen, the hydrocracking process is a large time-delay system with complex physicochemical reactions and long-processing technology in a series of devices or apparatuses. It often costs hours or more time to obtain the product from the feedstocks. Usually, quality attributes, such as the initial boiling points of the heavy naphtha and aviation kerosene, are key performance indicators (KPIs) that can reflect the sufficiency and efficiency of the process. The timely measurement of these KPIs can provide real-time feedback for process monitoring, control, and optimization. However, these key variables are often very difficult to measure online. Moreover, since the reaction kinetics are very complex, a first-principles model cannot meet the industrial estimation requirements due to process variations like the changes in feed compositions, operating conditions, and catalyst deactivation. Hence, most of the KPIs are obtained using offline laboratory tests, which results in large time delay for process control and monitoring.

To deal with this problem, soft sensors have been adopted to predict the KPIs in this process based on historical process data. The initial boiling points of the heavy naphtha and aviation kerosene are selected as the quality variables to be predicted in this study. For this purpose, 43 process variables like temperature and pressure were chosen as secondary variables for the soft sensors [46].

B. Prediction for the Initial Boiling Point of Heavy Naphtha

The initial boiling point of heavy naphtha is the temperature at the end of the condenser when the first drop is obtained. The data were collected from a real-industrial refinery in China, for a period of about two years from December 20, 2016 to September 29, 2018 with the sampling frequency of one quality sample every day. For confidentiality reasons, all the variables are normalized to the range 0 to 1. To build the soft sensor model, 650 labeled samples are collected from this process. The first 350 samples are selected as the training data, the middle 100 samples are used for model parameter validation, and the remaining 200 are used for the testing dataset.

For STA-LSTM in this study, the forecasting length is one sample for each subwindow. That is to say, once one sample is predicted, the window is moved forward by one step to predict the next sample. First, some of the main parameters that need to be optimized in the model are the number of time steps in the subwindow T; the number of neurons p in the hidden layer for the encoder LSTM; and the number of neurons q in the hidden layer for the decoder LSTM. The window size T is selected with a grid search from the candidate set $\{2, 3, 4, 5, 6\}$. Detailed RMSE for the window size is shown in Fig. 6. As can be seen from Fig. 6, T is selected to be three in this experiment since this achieves the best performance on the validation set for the model. For simplicity, p is set to be equal to q. As well, these two parameters are determined using a grid search on the set $\{15, 30, 60, 90, 120\}$. When $p = q = 60$, the model achieves the best performance for the validation set. Thus, the number of neurons is selected as 60 in this study. Next, the prediction performance of RMSE as a function of different window size is investigated. As well, the minibatch size, which is the number

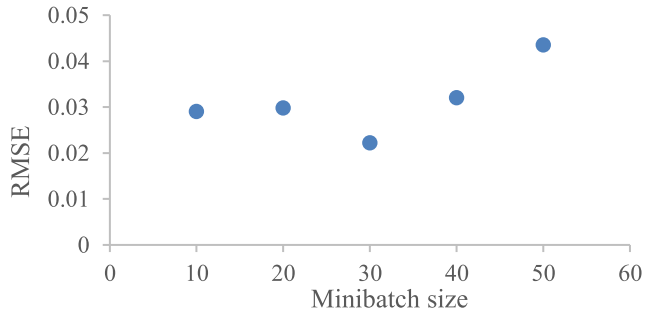

Fig. 7. Relationship between RMSE and minibatch size for STA-LSTM for the initial boiling point of heavy naphtha dataset.

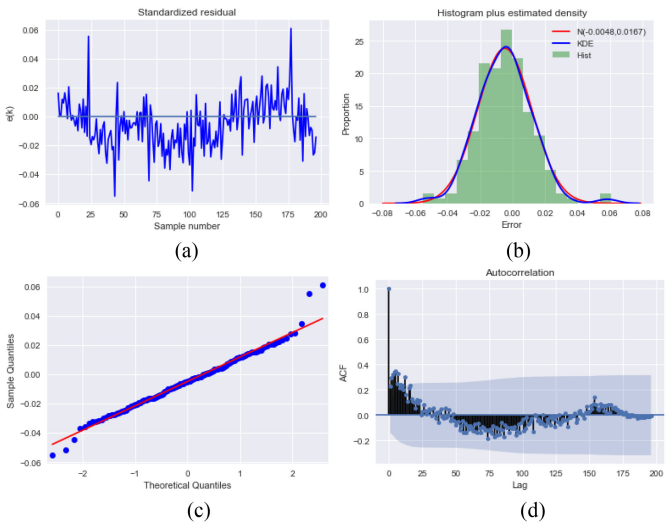

Fig. 8. Model validity tests for STA-LSTM for the prediction of the initial boiling point of heavy naphtha. (a) Model residuals. (b) Histogram of the residuals. (c) Quantile-quantile plot. (d) Residual autocorrelation.

of samples in each minibatch, should be determined for model training. By changing the minibatch size from the set {10, 20, 30, 40, 50}, the corresponding prediction performance of RMSE is obtained using the validation dataset. Fig. 7 shows the results. It can be seen that the optimal minibatch size is 30. Similarly, the learning rate is set as 0.001 and the epoch number is determined as 100 during the training process in this case. In addition, the neural network in spatial attention and temporal attention are both two-layer neural networks. Thus, the number of input and output neurons in the SA is equal to the dimension of the input variables. For the TA, the number of input and output are the window size T and the dimension of predicted quality variable, respectively.

It takes 31.99 s to train the proposed STA-LSTM-based soft sensor model and 0.14 s for testing of the initial boiling point of heavy naphtha. It is essential to validate the structure and parameters for the proposed model [47], [48]. Hence, some statistical tests, like the detailed prediction residuals, the histogram, the quantile–quantile plot and residual autocorrelation, are carried on the model residuals, which are shown in Fig. 8. Fig. 8(a) shows the detailed prediction errors for each testing sample.

TABLE I
PREDICTION RMSE OF THE THREE METHODS FOR THE INITIAL BOILING POINT OF HEAVY NAPHTHA ON THE TESTING DATASET

Method	$\text{RMSE}_{\text{testing}}$	$R^2_{testing}$
ARIMAX	0.0767	0.3031
DBN	0.0754	0.3262
RNN	0.0521	0.4162
LSTM	0.0574	0.6092
Attention LSTM	0.0397	0.8082
STA-LSTM	0.0184	0.9548

Fig. 8(b) and (c) shows that the prediction errors are normally distributed. Fig. 8(d) gives the residual autocorrelations, which shows that the residuals are as required independent of each other. Hence, the prediction errors are normally distributed and independent, which indicates that the proposed model is good.

Table I compares the prediction performance for the six soft sensors using the testing dataset. The moving average order is optimized as three and the autoregressive order is determined as four for ARIMAX. From Table I, ARIMAX has the worst prediction performance on the testing dataset. Since it is a linear model, it is difficult to capture the nonlinear relationships of this process. For the DBN, it can model the nonlinear relationship in process data. Thus, it performs better than ARIMAX. However, it is a static method that the data temporal relationship is not taken into consideration for modeling. The RNN has the ability of capturing the dynamic nature in process data. Thus, RNN outperforms DBN. However, it is difficult for RNN to learn long-term dependencies. For the LSTM network, it can partially extract the nonlinear characteristics of the data through the nonlinear activation function and use a memory cell to store long-term information for quality variable prediction. However, it does not give different attention to the subwindow data at different time steps. Hence, it may lose important hidden state information from the past. On the other hand, attention-based LSTM model uses an attention mechanism to take the previous decoder hidden state as a reference. Then, different attention weights are assigned to the encoder hidden states across the time steps in the subwindow for prediction of each query data. In this way, attention-based LSTM can get much better prediction performance than LSTM. For the proposed STA-LSTM model, the spatiotemporal mechanism can, not only adaptively discover the relevant samples across the time steps in the window, but also adaptively identify the input variables related to the quality variable at the current time step t. Hence, the proposed method has better prediction accuracy since the spatial attention mechanism and temporal attention mechanism are simultaneously introduced for adaptive modeling. Furthermore, the detailed predictions on the testing dataset are shown in Fig. 9 for ARIMAX, DBN, RNN, LSTM, attention-based LSTM and STA-LSTM. As can be seen from Fig. 9, the prediction performances of the ARIMAX, DBN, RNN, and LSTM-based soft sensors are the worst, since their prediction curves do follow the measured values. For attention-based LSTM, the prediction curve tracks the changes in measured values much better, but there are still large deviations between the predicted and measured output

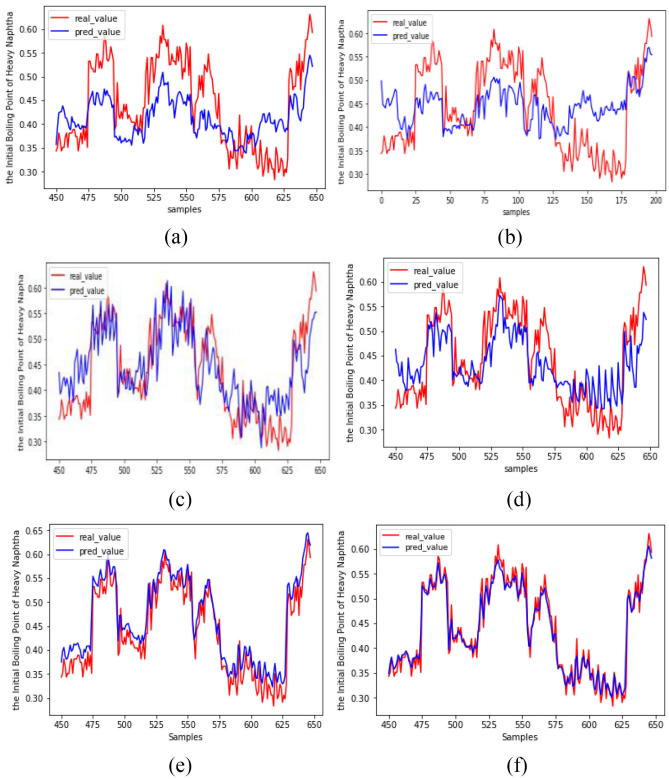

Fig. 9. Detailed prediction results for the initial boiling point of heavy naphtha: (a) ARIMAX. (b) DBN. (c) RNN. (d) LSTM. (e) Attention-based LSTM. (f) STA-LSTM.

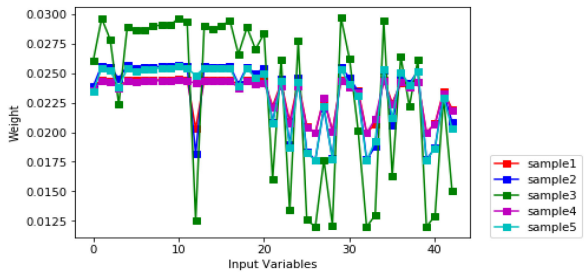

Fig. 10. Spatial variable attention weights of five samples in STA-LSTM for the initial boiling point of heavy naphtha.

values. However, by introducing the spatial and temporal attention mechanism into the attention-based LSTM, the prediction curves can track the measured output curve very well. Fig. 10 shows the spatial variable attention weights of five samples for the STA-LSTM model in the testing set. It can be seen that the input variables have different importance for the prediction and the quality variable.

C. Prediction for the Initial Boiling Point of Aviation Kerosene

Furthermore, the proposed STA-LSTM model is applied to predicting the initial boiling point of the aviation kerosene. The dataset was collected from January 19, 2016 to November 30, 2018 with a sampling frequency of 12 h per sample. A total of

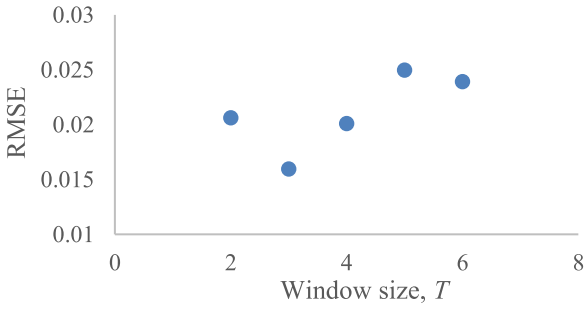

Fig. 11. Relationship between RMSE and window size, T, for STA-LSTM using the initial boiling point of aviation kerosene dataset.

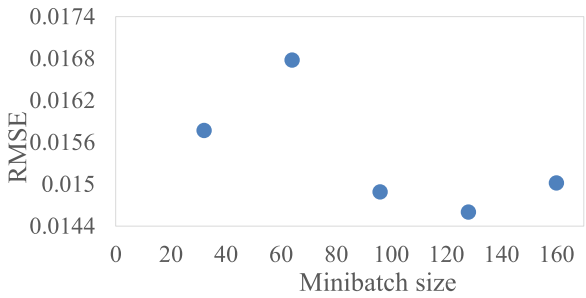

Fig. 12. Relationship between RMSE and minibatch size for STA-LSTM for the initial boiling point of aviation kerosene dataset.

1689 samples were collected. To build the soft sensor model, the first 800 samples are selected as the training data. The next 200 samples are used for model parameter validation and the remaining 689 are for testing purposes. As before, the process and quality variables are standardized before modeling.

Also, the important parameters should be determined for the models. The first parameter is the window size T. Similarly, by changing the window size from the candidate set $\{2, 3, 4, 5, 6\}$, the prediction RMSE on the validation set is obtained after model training. Detailed results are given in Fig. 11. As can be seen from Fig. 11, the model achieves the best performance using the validation set when the window size is $T = 3$. Thus, the window size T is set to be 3 for STA-LSTM. Similarly, the number of neurons is selected as 60 for the encoder and decoder. Then, the minibatch size is investigated again in this case, which is selected from the set $\{32, 64, 96, 128, 160\}$. Here, the prediction performance of RMSE with THE minibatch is calculated for the validation set. The results are shown in Fig. 12. As can be seen, the validation prediction error reaches a minimum when the minibatch size is 128. Hence, the minibatch size is set to be 128. In addition, the learning rate is determined as 0.01. Also, the epoch number is set to 120, since the value of loss function converges after 120 training iterations.

It takes 56.49 s to train the proposed STA-LSTM-based soft sensor model and 0.47 s for testing. Also, some tests are performed on the model residuals, which are shown in Fig. 13. It can be seen from Fig. 13(a) that the prediction errors are randomly distributed. Fig. 13(b) and (c) shows that the prediction errors are approximately normally distributed. Fig. 13(d) shows the

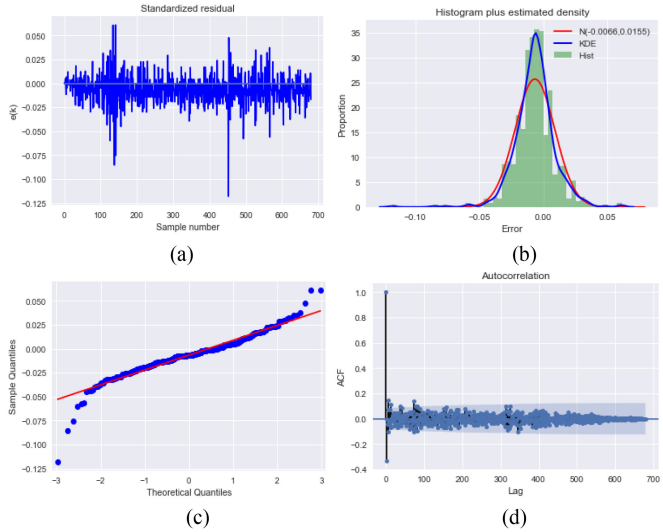

Fig. 13. Model validity tests for STA-LSTM for the prediction of the initial boiling point of aviation kerosene. (a) Model residuals. (b) Histogram of the residuals. (c) Quantile–quantile plot. (d) Residual autocorrelation.

TABLE II
PREDICTION RMSE OF THREE METHODS FOR THE INITIAL BOILING POINT OF AVIATION KEROSENE USING THE TESTING DATASET

Method	$\text{RMSE}_{testing}$	$R^2_{testing}$
ARIMAX	0.6420	0.0508
DBN	0.6326	0.0565
RNN	0.6011	0.1086
LSTM	0.5969	0.1820
Attention LSTM	0.4624	0.5091
STA-LSTM	0.2620	0.7613

residual autocorrelations, which suggest that the residuals are independent of each other. Thus, STA-LSTM has a good model structure and parameters with accurate prediction performance.

Similarly, ARIMAX, DBN, RNN, LSTM, attention-based LSTM and STA-LSTM are used to predict the initial boiling point of aviation kerosene. Table II gives the prediction results of the six methods on the testing dataset. As before, the ARIMAX, DBN, RNN, and LSTM networks give the worst prediction accuracy, while the attention-based LSTM provides a better prediction by using an attention mechanism to take the previous hidden states into consideration and assign different attention to them across the time steps. However, STA-LSTM provides the best prediction performance, since it can incorporate the most relevant information into its prediction. Fig. 14 further shows the detailed prediction values for the testing dataset with ARIMAX, DBN, RNN, LSTM, attention-based LSTM and STA-LSTM. From Fig. 14, the prediction of the proposed STA-LSTM method is in good match with the actual trajectory of the initial boiling point of aviation kerosene, and thus has a much smaller deviation. It can easily be seen that STA-LSTM provides the best prediction curve among the six methods. Also, Fig. 15 shows the spatial attention weight of each input variable for five samples, from which it can be seen that some variables are more important in predicting the quality of this sample.

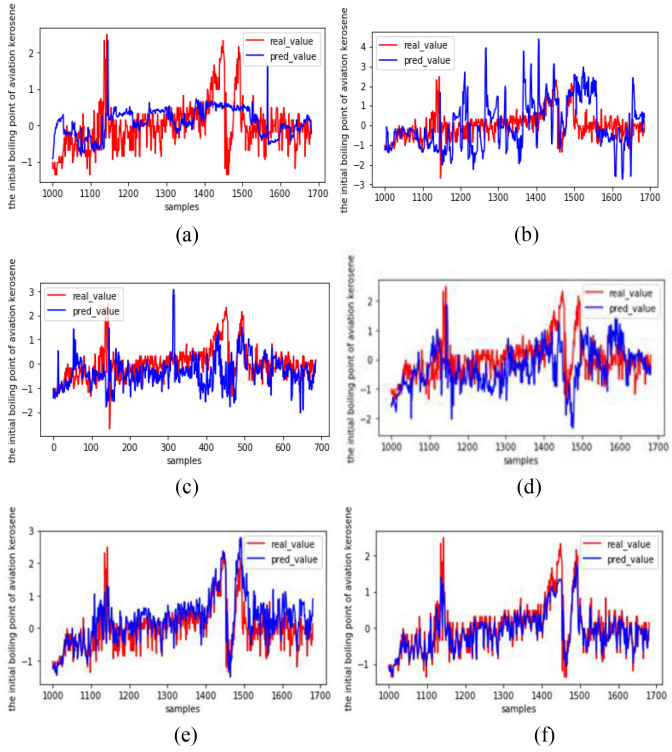

Fig. 14. Prediction performance for the initial boiling point of aviation kerosene: (a) ARIMAX. (b) DBN. (c) RNN. (d) LSTM. (e) Attention-based LSTM. (f) STA-LSTM.

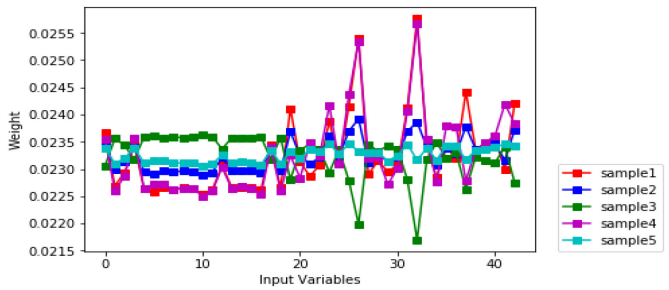

Fig. 15. Spatial variable attention weights of five samples in STA-LATM for the initial boiling point of aviation kerosene.

V. CONCLUSION

In this article, an attention-based framework was introduced for data-driven soft sensor modeling of industrial data time series. Since traditional attention-based LSTM networks only focus on the adaptive selection of the hidden states at different time steps, the importance of input variables with quality prediction was not considered for attention modeling. Hence, a new spatiotemporal attention-based LSTM model was proposed to obtain both the spatial variable and temporal sample attention for accurate modeling. A spatial attention mechanism was used to adaptively discover the input variables that are related to the quality variable and the attention weights are given for each input variable. Then, the temporal attention mechanism was used to model the temporal relevance of the hidden states at different time steps. Finally, the proposed STA-LSTM model

was applied to an industrial hydrocracking process for quality prediction. The results showed that the proposed model outperforms the ARIMAX, DBN, RNN, LSTM, and attention-based LSTM models. As can be seen, these models require the process data to be uniformly sampled with a fixed frequency. Future work will focus on modeling with an irregular sampling rate and providing multistep predictions for industrial processes.

REFERENCES

[1] X. Yuan, Y. Wang, C. Yang, Z. Ge, Z. Song, and W. Gui, "Weighted linear dynamic system for feature representation and soft sensor application in nonlinear dynamic industrial processes," *IEEE. Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1508–1517, Feb. 2018.

[2] S. Khatibisepehr, B. Huang, and S. Khare, "Design of inferential sensors in the process industry: A review of Bayesian methods," *J. Process. Control.*, vol. 23, no. 10, pp. 1575–1596, 2013.

[3] X. Yuan, C. Ou, Y. Wang, C. Yang, and W. Gui, "A layer-wise data augmentation strategy for deep learning networks and its soft sensor application in an industrial hydrocracking process," *IEEE Trans. Neural. Net. Lear. Syst.*, to be published, doi: 10.1109/TNNLS.2019.2951708.

[4] M. Järvisalo, T. Ahonen, J. Ahola, A. Kosonen, and M. Niemelä, "Soft-sensor-based flow rate and specific energy estimation of industrial variable-speed-driven twin rotary screw compressor," *IEEE. Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3282–3289, May 2016.

[5] 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.

[6] 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. T.*, vol. 25, no. 3, pp. 1124–1132, May 2017.

[7] W. Shao and X. Tian, "Adaptive soft sensor for quality prediction of chemical processes based on selective ensemble of local partial least squares models," *Chem. Eng. Res. Des.*, vol. 95, pp. 113–132, 2015.

[8] 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," *ISAT.*, vol. 98, pp. 403–417, 2020.

[9] Y. A. Shardt, *Statistics for Chemical and Process Engineers: A Modern Approach*, Cham, Switzerland: Springer International Publishing, 2015.

[10] 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.

[11] 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.

[12] J. Zheng, Z. Song, and Z. Ge, "Probabilistic learning of partial least squares regression model: Theory and industrial applications," *Chemometr. Intell. Lab. Syst.*, vol. 158, pp. 80–90, 2016.

[13] X. Yuan, J. Zhou, and Y. Wang, "A spatial-temporal LWPLS for adaptive soft sensor modeling and its application for an industrial hydrocracking process," *Chemometr. Intell. Lab. Syst*, vol. 197, 2020, Art. no. 103921.

[14] Y. Wang, D. Wu, and X. Yuan, "A two-layer ensemble learning framework for data-driven soft sensor of the diesel attributes in an industrial hydrocracking process," *J. Chemometr.*, vol. 33, no. 12, 2019, Art. no. e3185.

[15] X. Yuan, S. Qi, and Y. Wang, "Stacked enhanced auto-encoder for data-driven soft sensing of quality variable," *IEEE Trans. Instrum. Meas.*, to be published, doi: 10.1109/TIM.2020.2985614.

[16] M. Dam and D. N. Saraf, "Design of neural networks using genetic algorithm for on-line property estimation of crude fractionator products," *Comput. Chem. Eng.*, vol. 30, no. 4, pp. 722–729, 2006.

[17] N. M. Ramli, M. A. Hussain, B. M. Jan, and B. J. N. Abdullah, "Composition prediction of a debutanizer column using equation based artificial neural network model," *Neurocomputing*, vol. 131, no. 12, pp. 59–76, 2014.

[18] G. E. Hinton, "Learning multiple a layers of representation," *Trends. Cogn. Sci.*, vol. 11, no. 10, pp. 428–434, 2007.

[19] 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, 2018.

[20] S. Chao, Y. Fan, 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, 2014.

[21] 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*, to be published, doi: 10.1016/j.neucom.2018.11.107.

[22] S. Graziani and M. G. Xibilia, "A deep learning based soft sensor for a sour water stripping plant," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, 2017, pp. 1–6.

[23] X. Yuan, Y. Gu, Y. Wang, C. Yang, and W. Gui, "A deep supervised learning framework for data-driven soft sensor modeling of industrial processes," *IEEE Trans. Neurual Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2019.2957366.

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

[25] X. Yuan, C. Ou, Y. Wang, C. Yang, and W. Gui, "A novel semi-supervised pre-training strategy for deep networks and its application for quality variable prediction in industrial processes," *Chem. Eng. Sci.*, vol. 217, 2020, Art. no. 115509.

[26] K. Wang, B. Gopaluni, J. Chen, and Z. Song, "Deep learning of complex batch process data and its application on quality prediction," *IEEE. Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2018.2880968.

[27] X. Yuan, Y. Wang, C. Yang, and W. Gui, "Stacked isomorphic autoencoder based soft analyzer and its application to sulfur recovery unit," *Inform. Sci.*, to be published.

[28] Y. Wang, Z. Pan, X. Yuan, C. Yang, and W. Gui, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," *ISA T.*, vol. 96, pp. 457–467, 2020.

[29] 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.

[30] 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.

[31] A. Graves, "Supervised sequence labelling with recurrent neural networks," *Studies Comput. Intell.*, vol. 385, pp. 5–13, 2008.

[32] 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.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural. Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] 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.

[35] X. Yuan, L. Li, Y. Wang, C. Yang, and W. Gui, "Deep learning for quality prediction of nonlinear dynamic process with variable attention-based long short-term memory network," *Can. J. Chem. Eng.*, to be published, doi: 10.1002/cjce.23665.

[36] G. Kurata, B. Ramabhadran, G. Saon, and A. Sethy, "Language modeling with highway LSTM," in *Proc. ASRU 2017*, Okinawa, Japan, 2017, pp. 244–251.

[37] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches," in *Proc. Int. Conf. Artif. Neural Netw.*, 2001, pp. 669–676.

[38] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Underst.*, 2013, pp. 273–278.

[39] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Comput. Sci.*, 2014, *arXiv:1409.0473*.

[40] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2627–2633.

[41] V. Mnih, N. Heess, and A. Graves, "Recurrent models of visual attention," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.

[42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[43] K. Cho, B. V. Merrienboer, D. Bahdanau, and Y. J. C. S. Bengio, "On the properties of neural machine translation: encoder-decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. Stat. Transl.*, 2014, pp. 103–111.

[44] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization[J], 2014, *arXiv:1412.6980*.
[45] X. Yuan, J. Zhou, Y. Wang, and C. Yang, "Multi-similarity measurement driven ensemble just-in-time learning for soft sensing of industrial processes," *J. Chemometr.*, vol. 32, no. 9, 2018, Art. no. e3040.
[46] X. Yuan, J. Zhou, and Y. Wang, "A comparative study of adaptive soft sensors for quality prediction in an industrial refining hydrocracking process," in *Proc. IEEE 7th Data Driven Control Learn. Syst. Conf.*, 2018, pp. 1064–1068.
[47] S. A. BlLlings and W. S. F. Voon, "Correlation based model validity tests for non-linear models," *Int. J. Control*, vol. 44, no. 1, pp. 235–244, 1986.
[48] S. A. Billings and W. S. F. Voon, "A prediction-error and stepwise-regression estimation algorithm for non-linear systems," *Int. J. Control*, vol. 44, no. 3, pp. 803–822, 1986.

Xiaofeng Yuan (Member, IEEE) received the B.Eng. and Ph.D. degrees in automation from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2011 and 2016, respectively.

He was a Visiting Scholar with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada, from November 2014 to May 2015. He is currently an Associate Professor with the School of Automation, Central south University, Changsha, China. His research interests include deep learning and artificial intelligence, machine learning and pattern recognition, industrial process soft sensor modeling, process data analysis, etc.

Lin Li received the B.Eng. degree in automation from the School of Information Engineering, Xiangtan University, Xiangtan, China, in 2018. She is now a Master Student in School of Automation, Central South University, Changsha, China.

Her research interests include deep learning, machine learning, soft sensor modeling, process data mining, etc.

Yuri A. W. Shardt received the doctoral degree under the supervision of Prof. B. Huang from the University of Alberta, Edmonton, AB, Canada, in 2012.

He is currently a Professor and Chair of the Department of Automation Engineering with the Technical University of Ilmenau, Ilmenau, Germany. Previously, he was an Assistant Professor with the University of Waterloo, Department of Chemical Engineering and a holder of the prestigious Alexander von Humboldt Scholarship with the University of Duisburg-Essen, Institute of Control and Complex Systems. He has written a book, entitled *Statistics for Chemical and Process Engineers: A Modern Approach* that focuses on the required mathematical background in order to implement advanced statistical methods using Excel and MATLAB. This book has been translated into German and is scheduled to be published in late 2020 by Springer as *Methoden der Statistik und Prozessanalyse*. He has written numerous papers that have appeared in such journals as *Automatica*, *Journal of Process Control*, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and *Industrial and Engineering Chemistry Research* and presented at multiple conferences. He has taught various courses in the intersection between statistics, chemical engineering, process control, EXCEL, and MATLAB. His thesis examined the methods for extracting valuable data for system identification from data historians for application to soft sensor design. In addition to his academic work, he has spent considerable time in industry working on implementing various process control solutions. He also has interests in linguistics, as well as software internationalisation and localization. His research interests include big data, system identification, data quality assessment, holistic control, and the smart world.

Yalin Wang (Member, IEEE) received the B.Eng. and Ph.D. degrees in control science and engineering from the Department of Control Science and Engineering, Central South University, Changsha, China, in 1995 and 2001, respectively.

Since 2003, she has been with the School of Information Science and Engineering, Central south University, where she was at first an Associate Professor and then a Professor. She is currently a Professor with the School of Automation, Central South University. Her research interests include the modeling, optimization and control for complex industrial processes, intelligent control, and process simulation.

Chunhua Yang (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 Central South University. Her current research interests include modeling and optimal control of complex industrial process, intelligent control system, and fault-tolerant computing of real-time systems.