Dynamic Soft Sensor Modeling And Its Application Using TCN-CBAM-LSTM Method

Jun Li* and Yang Hao

School of Automation and Electrical Engineering, Lanzhou Jiaotong University, Lanzhou, Gansu 730070, P.R. China

Corresponding author. E-mail: lijun691201@mail.lzjtu.cn

Received: July 10, 2024; Accepted: February 01, 2025

Data-driven soft sensor modeling has gained significant traction in modern chemical process monitoring and quality prediction. However, persistent challenges remain in accurately characterizing the complex dynamics inherent in chemical production systems, which typically exhibit significant time delays, strong nonlinearity, and time-varying characteristics. To address these critical challenges, a dynamic soft sensor modeling method based on temporal convolutional network (TCN) combined with channel spatiotemporal attention module and long short-term memory network (TCN-CBAM-LSTM) is proposed. Firstly, TCN is employed to extract deep nonlinear dynamic dependencies from process variables through its dilated causal convolution architecture, secondly, a convolutional block attention module (CBAM) is incorporated to enhance feature representation by adaptively focusing on critical spatiotemporal information across different sensor channels, finally, a long short-term memory network (LSTM) is integrated to model intricate temporal patterns and long-range dependencies between process variables and quality indicators. This multi-stage architecture enables comprehensive learning of both local temporal features and global dynamic relationships within complex chemical processes. To verify the effectiveness of the proposed method, TCN-CBAM-LSTM was applied to a soft sensor modeling example for calculating the exhaust gas composition in a sulfur recovery unit (SRU). Under the same experimental conditions, it was also compared with convolutional neural network (CNN), variable weighted stacked autoencoder (VW-SAE), spatiotemporal attention LSTM (STA-LSTM), CNN-LSTM, TCN, and TCN-LSTM. The results show that the TCN-CBAM-LSTM method has better performance and modeling accuracy, and its performance meets the needs of practical engineering applications.

Keywords: Soft sensor; Dynamic modeling; Temporal convolutional network; Convolutional block attention module; Long short-term memory network

http://dx.doi.org/10.6180/jase.202512_28(12).0003

1. Introduction

In the last few years, the monitoring of product quality standards and environmental pollution in actual chemical production processes has garnered increasingly widespread attention [1]. However, under current technical conditions, the high cost of online measuring instruments and the sometimes time-delay response can hinder the realization of optimized operations in process automation. Soft sensor modeling techniques can efficiently resolve this problem. The "soft instrument" formed by soft sensor modeling has the advantages of low cost and adaptability. It provides an inexpensive and rapid estimation method for the primary variables that need to be monitored in the process, making it possible to design effective and fault-tolerant control strategies.

At present, data-driven soft sensor modeling techniques have been extensively implemented. Classic data-driven

soft sensor methods primarily use artificial intelligence techniques such as feed-forward neural network [2] and support vector machines (SVM) [3]. However, due to their shallow learning architecture, it has become challenging for these methods to meet the demands as the scale of the sample size increases. Compared to shallow learning methods, deep learning methods have significant advantages in training with large-scale datasets. Deep learning techniques, including autoencoders (AE), recurrent neural networks (RNN), and CNN, have been used in soft sensor modeling. For instance, de Castro-Cros et al [4]. used an autoencoder architecture to establish a soft sensor model that can monitor and evaluate the state of a specific industrial gas turbine compressor; Devakumar et al [5]. proposed a LSTM-based soft sensor model (LSTM-SS) that can effectively calculate three key quality parameters of liquid rocket engines (LREs); Abad et al [6]. used CNN to effectively predict oil flow rates through orifice plates based on sensor parameter measurement data.

TCN uses methods like causal dilated convolutions and residual connections to get a greater receptive field, allowing it to capture long-term historical information. The residual connection technique helps prevent issues like "vanishing gradients" and "exploding gradients" within the model. TCN has been used in a number of areas, like motion sensor data prediction [7], stock price prediction [8], renewable energy forecasting [9], and chaotic time series prediction [10]. For example, Yuan et al [11]. proposed an autoregressive TCN (AR-TCN) method, which achieved good results in product quality prediction for industrial dehydrogenation units and hydrocracking processes; Sun et al [12]. proposed a soft measurement model using TCN and Attention Simple Recurrent Unit (ASRU) networks, which has been applied in modeling the specific surface area of cement.

On the other hand, incorporating CBAM which contains Channel and Spatial Attention Modules into deep learning models can further enhance the modeling capabilities of existing soft measurement models. For instance, Cheng et al [13]. integrated CBAM into TCN for chaotic time series prediction and achieved good results. Additionally, to further improve model performance, some researchers have started combining different deep learning models. Examples include the hybrid model combining CNN and LSTM (CNN-LSTM by Nguyen et al [14].) and the combination of TCN and LSTM (TCN-LSTM by Ren et al [15].). These combined models have also achieved impressive modeling results.

Considering the dynamic characteristics of chemical processes and the advantages of TCN in handling high-dimensional complex problems, this paper proposes a TCN-CBAM-LSTM method. The TCN is used to extract nonlinear and long-term dynamic dependency features from process variables. The CBAM is employed to calibrate the input feature maps, focusing on useful information, while the LSTM network captures the dynamic dependencies between inputs and outputs, thereby addressing the delay issues between variables to some extent. In order to verify the validity of the TCN-CBAM-LSTM method, it was used to calculate the tail gas composition of SRU. Under the same conditions, this method was compared with CNN, CNN-LSTM, STA-LSTM, VW-SAE, TCN, and TCN-LSTM. The experimental results demonstrate that the proposed model exhibits satisfactory performance and potential for application.

The remainder of the paper is organized as follows: section 2 describes the structure of TCN-CBAM-LSTM; section 3 experimentally verifies the performance of TCN-CBAM-LSTM; and section 4 sums up the conclusions of this study.

2. Tcn-cbam-lstm model

2.1. TCN

TCN is an innovative deep learning model derived from traditional CNN. It incorporates core techniques of classical CNN, including local connectivity, weight sharing, and feature extraction. TCN incorporates causal convolutions, dilated convolutions, and residual connections, endowing it with high parallel computing capabilities and flexible receptive fields, thus granting it stronger historical memory capabilities compared to RNNs. This section will specifically introduce the components of TCN.

2.1.1. Causal Dilated Convolution

In order to predict without involving future information, TCN uses a causal convolution strategy. The causal convolution is computed using the input features at the time t and before the time t for the convolution. For a given sequence data $\boldsymbol{X} = (x_0, x_1, \cdots, x_T)$ and convolution kernel $\boldsymbol{F} = (f_0, f_1, \cdots, f_{K-1})$, the causal convolution at time t is defined as:

$$(\boldsymbol{F} \times \boldsymbol{X})(t) = \sum_{k=0}^{K-1} f_k x_{t-(K-1-k)} \tag{1}$$

where K is the size of the convolution kernel and $x_{t-(K-1-k)}$ prevents future sequence information $x_{t+1}, x_{t+2}, \cdots, x_{t+n}$ from participating in the convolution operation.

Furthermore, the dilated convolution obtains a larger sensory field by spaced sampling. This allows the receptive field size to grow exponentially without altering the original model parameters. Combining causal and dilated

convolutions allows us to analyze past data while expanding the sensory reach. Causal dilated convolution at time t can be formulated as:

$$(\boldsymbol{F} \times \boldsymbol{X})(t) = \sum_{k=0}^{K-1} f_k x_{t-d(K-1-k)} \tag{2}$$

where d is the dilated ratio. The expansion rate of layer i is 2^{i-1}. In addition, a zero-padding mechanism is used to complement the sequence to ensure that the convolution can be performed correctly.

2.1.2. *Residual Connection*

When using causal dilated convolution, the convolution kernel can have a large sensory field. However, the dilated ratio d grows exponentially with the network depth, causing the network to degrade. To alleviate this problem, a deep residual learning method is used, where an additional residual connection is added to construct the multilayer network, and the input x is added directly to the output y. The mathematical expression is:

$$y = f(x; \boldsymbol{W}) + x \tag{3}$$

where y is the output, and $f(x; \boldsymbol{W})$ represents the residual feature.

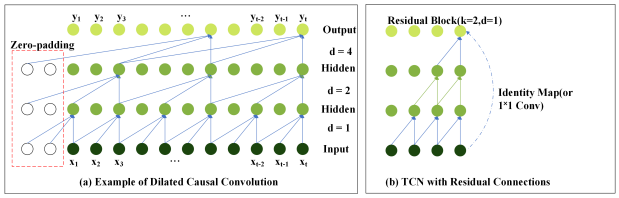

Fig. 1. An example of causal dilated convolution and residual connection.

TCN taking causal dilated convolution is shown in Fig. 1(a). As illustrated in the accompanying figure, the output is generated by both the preceding inputs and does not utilize any future information.

As illustrated in Fig. 1(b), TCN employs residual connections to create a deep network with a larger receptive field. These connections link the initial and final layers. Optional 1-dimensional convolution to ensure that the input and output shapes are the same.

2.2. CBAM

CBAM consists of two modules, channel attention module (CAM) and spatial attention module (SAM), and implements a sequential attention structure from channel to space. Given an input feature $\boldsymbol{F} \in \mathbb{R}^{C \times H \times W}$, one-dimensional CAM $\boldsymbol{F} \in \mathbb{R}^{C \times 1 \times 1}$, and two-dimensional SAM $\boldsymbol{M_s} \in \mathbb{R}^{1 \times H \times W}$, the complete CBAM process can be depicted as:

$$\begin{aligned} \boldsymbol{F'} &= \boldsymbol{M_c}(\mathbf{F}) \otimes \boldsymbol{F}, \\ \boldsymbol{F''} &= \boldsymbol{M_s}\left(\boldsymbol{F'}\right) \otimes \boldsymbol{F'} \end{aligned} \tag{4}$$

where $\otimes$ denotes an element-by-element multiplication and $\boldsymbol{F''}$ is the final output. Fig. 2 illustrates how CBAM works.

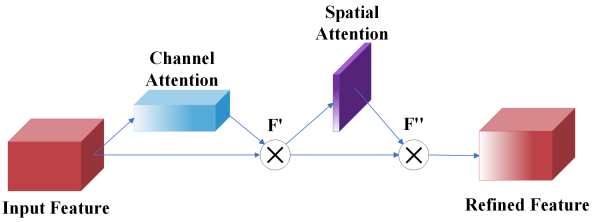

Fig. 2. Overview of CBAM.

The composition of the two attention modules is described in detail below. First, two feature descriptions $\boldsymbol{F^c_{avg}}$ and $\boldsymbol{F^c_{max}}$ of different dimensions are obtained by integrating the spatial information of the input features using global average pooling and global maximum pooling operations. The two pooled features share a multilayer perceptron network. Initially, compression and recovery of channel counts through multiple fully connected layers. After a sigmoid activation function normalizes the channel weights to between 0 and 1 , the two feature maps are stacked in the channel dimension. Finally, the input feature map is multiplied by these normalized weights.

The channel attention is calculated as follows:

$$\begin{aligned} \boldsymbol{M_c}(\boldsymbol{F}) &= \sigma(\mathrm{MLP}(\mathrm{Avg\,Pool}(\boldsymbol{F})) + \mathrm{MLP}(\mathrm{MaxPool}(\boldsymbol{F}))) \\ &= \sigma\left(\boldsymbol{W_1}\left(\boldsymbol{W}_0\left(\boldsymbol{F^c_{avg}}\right)\right) + \boldsymbol{W_1}\left(\boldsymbol{W_0}\left(\boldsymbol{F^c_{max}}\right)\right)\right) \end{aligned} \tag{5}$$

where σ denotes the sigmoid function, the multilayer perceptron weights $\boldsymbol{W}_0 \in \mathbb{R}^{C/\times C}$ and $\boldsymbol{W_1} \in \mathbb{R}^{C \times C/r}$ are shared for both inputs, the ReLU activation function follows $\boldsymbol{W_0}$, and $\boldsymbol{F}$ is the input feature.

Spatial attention focuses on attending to the spatial domain portion of information and complements channel attention. To compute spatial attention, average pooling and maximum pooling operations are first applied along the channel axis. The resulting pooled feature maps, $\boldsymbol{F^{ss}_{avg}} \in \mathbb{R}^{1 \times H \times W}$ and $\boldsymbol{F'_{max}} \in \mathbb{R}^{1 \times H \times W}$, are then stacked in the channel dimension. Then the channel information is fused using a convolutional kernel of 7×7 size, and a convolutional layer is applied to generate the spatial attention map $\boldsymbol{M_s}\left(\boldsymbol{F'}\right) \in \boldsymbol{R}^{H \times W}$. Finally, the convolutional result is normalized by applying spatial weights to the feature maps using a sigmoid function, and the input feature maps are then multiplied by these weights.

Spatial attention is calculated as follows:

$$\begin{aligned} \boldsymbol{M}_{\mathrm{s}}\left(\boldsymbol{F}'\right) &= \sigma\left(f^{7\times7}\left[\mathrm{AvgPool}\left(\boldsymbol{F}'\right);\mathrm{MaxPool}\left(\boldsymbol{F}'\right)\right]\right) \\ &= \sigma\left(f^{7\times7}\left(\left[\boldsymbol{F}^{\mathrm{s.}}_{\mathrm{avg}};\boldsymbol{F}^{\mathrm{s.}}_{\mathrm{max}}\right]\right)\right) \end{aligned} \tag{6}$$

where σ is a sigmoid function, represents a 7×7 convolution operation.

2.3. LSTM

LSTM, as an improved version of recurrent neural networks, have proven their effectiveness in processing series data in many applications.

LSTM utilizes memory cells with self-connections to store temporal states. It uses different status gates to maintain, modify, or delete unit status information. When the input sequence is x_t, the output h_t of the network is calculated using the following formula:

$$f_t = \sigma\left(\boldsymbol{W}_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{7}$$

$$i_t = \sigma\left(\boldsymbol{W}_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{8}$$

$$o_t = \sigma\left(\boldsymbol{W}_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{9}$$

$$\hat{c}_t = \tanh\left(\boldsymbol{W}_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{10}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t \tag{11}$$

$$h_t = o_t \cdot \tanh\left(c_t\right) \tag{12}$$

where , $\boldsymbol{W}_f, \boldsymbol{W}_i, \boldsymbol{W}_o, b_f, b_i$ and b_o are the weight matrices and the bias matrices of forget gate, input gate and output gate. σ is the sigmoid activation function. The input and forgetting gates determine the proportion of information that is accounted for by c_{t-1} and $\hat{c}_t$ in the current cell state c_t, where $\boldsymbol{W}_c$ and b_c are the weight and bias matrices.

2.4. TCN-CBAM-LSTM

The TCN-CBAM-LSTM based soft sensor model is shown in Fig. 3.

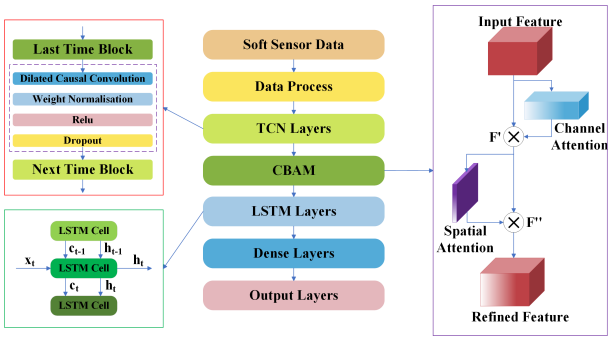

Fig. 3. TCN-CBAM-LSTM based soft sensor model.

It is divided into five main parts: First, preprocess and normalize the original sample set. Second, input the processed samples into TCN to extract time series features related to the output variables. Third, feed the features extracted by TCN into CBAM to emphasize important features and enhance focus on crucial information. Fourth, input the CBAM-processed information into LSTM to capture dynamic temporal relationships. Fifth, pass the data through a dense layers for output.

3. Soft sensor modeling examples for chemical process

To test the effectiveness of the TCN-CBAM-LSTM network in soft sensor, this section applies the model to two examples: predicting the H_2 S and SO_2 content in SRU tail gas. The hardware environment for the soft sensor models constructed, trained, and tested in this paper is an Intel(R) Core(TM) i7-12700H CPU @ 2.30 GHz processor, 16.0GB RAM, NVIDIA GeForce RTX 3060 Laptop GPU, and Windows 11 system. The simulation uses the Keras library in the TensorFlow environment.

To evaluate the performance of different models, root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and R-square (R^2) were selected as evaluation metrics:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{13}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{14}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{|y_i|}\left|y_i - \hat{y}_i\right|\right) \times 100\% \tag{15}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}_i\right)^2} \tag{16}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value of the model, n is the length of the sequence of target variables, and $\bar{y}_i$ is the average value of y_i.

3.1. Soft Sensing Modeling of H_2S and SO_2 Concentrations in SRU

As an important processing unit in a refinery, SRU is a device that recovers sulfuric acid-containing gases. The SRU reduces acid gas emissions and recovers elemental sulfur as a valuable by-product. The SRU primarily consists of four identical subunits, known as sulfur lines, each capable of converting acid gases into sulfur. Because hydrogen sulfide and sulfur dioxide often damage sensors and require frequent maintenance, establishing a soft sensor model is appropriate. In this study, one of the production lines was selected for soft sensor modeling. A simplified illustration of the SRU process is shown in Fig. 4.

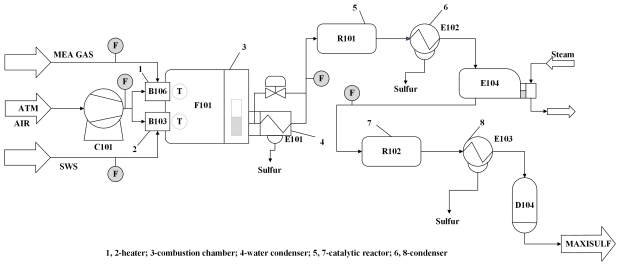

Fig. 4. Simplified scheme of SRU.

A detailed description of the dynamic characteristics of SRU can be found in reference [16]. The primary variables selected are the concentrations of H_2 S and SO_2. Based on expert knowledge and control requirements, five auxiliary variables are selected as modeling inputs, specifically:

x_1 : Gas flow rate;

x_2 : Air flow rate;

x_3 : Secondary air flow rate;

x_4 : Gas flow rate in the SWS area;

x_5 : Air flow rate in the SWS area.

The auxiliary variables are in m^3/h (cubic meters per hour) and the sampling period is 1 minute.

In the literature [16], a nonlinear moving average (NMA) model is used for the soft sensor model of the SRU device. To capture the historical information of key variables and further enhance accuracy, this paper adopts the following dynamic model:

$$y_1(k) = f_1\left(x_1(k), x_1(k-5), x_1(k-7), x_1(k-9), \cdots, x_5(k), x_5(k-5), x_5(k-7), x_5(k-9), y_1(k-1)\right) \quad (17)$$

$$y_2(k) = f_2\left(x_1(k), x_1(k-5), x_1(k-7), x_1(k-9), \cdots, x_5(k), x_5(k-5), x_5(k-7), x_5(k-9), y_2(k-1)\right) \quad (18)$$

where $y_1(k)$ and $y_2(k)$ represent the concentration output of H_2 S and SO_2 at time k.

In order to evaluate the performance of the proposed method, TCN-CBAM-LSTM and the compared CNNs, VW-SAE [17], STA-LSTM [18], CNN-LSTM, TCN, and TCN-LSTM are taken with fixed parameter settings in the experiments in this paper. The trial-and-error method is used to determine the optimal hyperparameters for the model. The deep learning models are trained using the Adam optimizer with a learning rate of 0.001, a batch size of 60, the ReLU activation function, and 100 iterations. The model hyperparameters of TCN-CBAM-LSTM mainly include: two TCN layers; the number of filters is 64 and 32, the size of convolutional kernel is 16; CBAM module: the channel compression ratio is 8, and the size of spatial convolutional kernel is 7×7; one LSTM layer: the number of units is 32; and two dense layers: neurons in each layer is 32 and 16, respectively. In order to facilitate comparisons, as well as to verify the effectiveness of TCN-CBAM-LSTM, CNN, CNN-LSTM, STA-LSTM, VW-SAE, TCN and TCN-LSTM were also constructed under the same conditions.

This experiment uses 10081 samples collected by this process, and the initial 8,000 samples were utilized as training sets, with a division ratio of approximately 0.8. The remaining samples were used as test data. Through the grid search method, it was determined that the input features of the samples were converted into a two-dimensional array for training and testing using a sliding window of size 9.

3.1.1. Soft sensor model of H_2S concentration in SRU

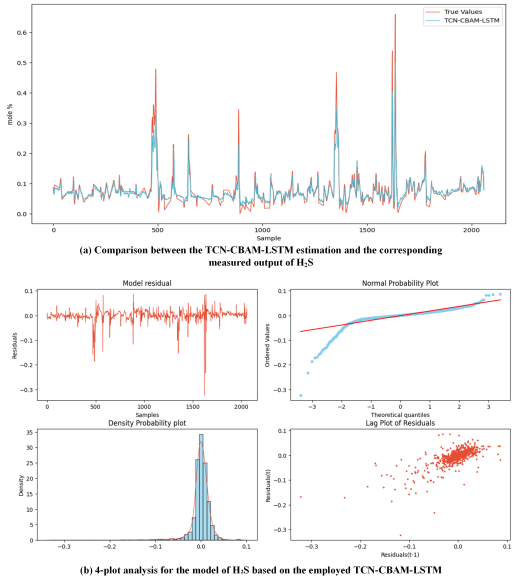

Fig. 5. H_2S concentration experiment analysis diagram.

Fig. 5(a) shows the comparison between the TCN-CBAM-LSTM estimation and the corresponding measured output of H_2S, where the red bars indicate the predicted values and the blue bars indicate the actual values. It can be seen that for the H_2S component content, TCN-CBAM-LSTM correctly predicted the trend. The four-plot analysis of the residuals of H_2S prediction by TCN-CBAM-LSTM is given in Fig. 5(b). From the four-plot analysis results, the TCN-CBAM-LSTM method has been demonstrated to be an effective model for this application. Table 1 gives indicators of models for estimating the output of H_2S.

The results of model performance comparison in Table 1 show that the TCN-CBAM-LSTM method achieves satisfactory accuracy. Further, the method of this paper is also compared with the results of the related literature, which [16] used four different methods for predictive estimation

Table 1. Indicators of models for estimating the output of H_2S.

Model	RMSE	MAE	MAPE(%)	R^2
CNN	0.03157	0.01855	1.652	0.6809
CNN-LSTM	0.02970	0.01665	1.483	0.7177
STA-LSTM	0.03030	0.01654	1.479	0.7061
VW-SAE	0.02954	0.01783	1.596	0.7206
TCN	0.02871	0.01620	1.441	0.7361
TCN-LSTM	0.02476	0.01278	1.136	0.8038
TCN-CBAM-LSTM	**0.02272**	**0.01244**	**1.112**	**0.8347**

Table 2. Indicators of models for predicting the content of SO_2.

Model	RMSE	MAE	MAPE(%)	R^2
CNN	0.03374	0.02422	2.075	0.6534
CNN-LSTM	0.02916	0.01957	1.658	0.7411
STA-LSTM	0.02434	0.01797	1.529	0.8196
VW-SAE	0.02826	0.02069	1.767	0.7568
TCN	0.02799	0.01855	1.581	0.7615
TCN-LSTM	0.02592	0.01788	1.534	0.7955
TCN-CBAM-LSTM	**0.02091**	**0.01384**	**1.176**	**0.8669**

of H_2S concentration based on the NMA model using multilayer perceptron (MLP), radial basis function neural network (RBF), adaptive neuro-fuzzy system (NFS), and nonlinear least squares (NLSQ), and the method with the best modeling estimation capability is the NLSQ dynamic soft sensor modeling method, with an RMSE of 0.02828 for H_2S on the test set. In summary, TCN-CBAM-LSTM is the best fitting model for the soft sensor model of H_2S component content.

3.1.2. Soft sensor model of SO_2 concentration in SRU

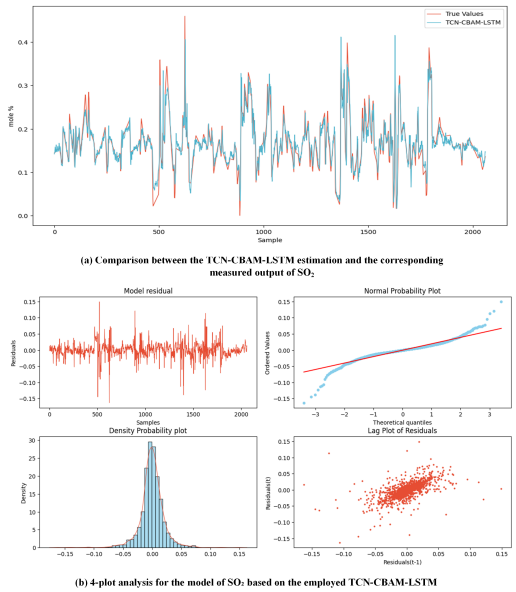

Fig. 6. SO_2 concentration experiment analysis diagram.

Fig. 6(a) shows the comparison between the TCN-CBAM-LSTM estimation and the corresponding measured output of SO_2, where the red bars indicate the predicted values and the blue bars indicate the actual values. It can be observed that the SO_2 component content was correctly predicted by TCN-CBAM-LSTM, which demonstrated the ability to identify trends. The four-plot analysis of the residuals of SO_2 prediction by TCN-CBAM-LSTM is given in Fig. 6(b). From the four-plot analysis results, it can be seen that TCN-CBAM-LSTM method is still a good choice of model. Table 2 gives a comparison of the results of different methods for predicting SO_2 concentration with respect to their errors on the test set.

The results of model performance comparison in Table 2 show that the TCN-CBAM-LSTM method still performs well in predicting SO_2. Further, the methodology of this paper is also compared with the results of related literature, In the field of industrial process modeling, a deep cascade generalized learning system with fast updating capability is proposed in the literature [19] for the purpose of modeling soft sensors, which has an RMSE of 0.0252 and an R^2 of 0.7697 when tested for SO_2 in SRU; In the literature [16] SO_2 was tested using NLSQ and the predicted performance index RMSE is 0.02. In summary, TCN-CBAM-LSTM is still the best fitting model for soft sensor of SO_2 content.

The model was proved to perform well through two experiments and was able to effectively and accurately predict the contents of H_2S and SO_2. From Figs. 5 and 6, it can be observed that the TCN-CBAM-LSTM method exhibits minimal prediction error drift across various points in the test dataset. The residual histogram displays an approximately

zero-mean, bell-shaped curve. Additionally, the normality plot approaches a straight line, indicating that the scatter of points shows no discernible correlation structure and minimal dispersion. These findings suggest that, in the soft sensor experiment of SRU tail gas component content, the modeling performance of TCN-CBAM-LSTM is satisfactory compared to other soft sensor modeling methods. According to the evaluation metrics, the TCN-CBAM-LSTM model outperforms existing competitive benchmarks. Compared to the TCN-LSTM, TCN-CBAM-LSTM predicted H_2S and SO_2 content with a reduction of 8.2% and 19.3% in RMSE, a reduction of 2.7% and 22.6% in MAE, and a reduction of 2.1% and 30.4% in MAPE, and an improvement in R2 of 3.8% and 9%, which is the best performance among existing competitive best performance in the existing competitive benchmarking.

4. Conclusions

Data-driven soft sensor models are highly adaptive, combining the advantages of low cost to provide an accurate and fast means of estimating process-dominant variables that need to be measured in real time. This paper introduces the TCN-CBAM-LSTM method. The deep features are extracted by TCN to learn the long-term historical dynamic trends of quality labels; the nonlinear mapping features from TCN are integrated and extracted by CBAM, which enhances the data representation in a generalized sense and allows for the extraction of fuller and more effective information from the raw data; the sequences are processed by LSTM layer, and finally integrated by a dense layer to obtain the output. The TCN-CBAM-LSTM model is applied to predict the concentrations of H_2S and SO_2 gases in the SRU. These examples are compared with existing methods and results from related literature. The experimental results confirm that the dynamic soft sensor method based on TCN-CBAM-LSTM achieves good approximation accuracy under the same conditions, which shows the effectiveness and application potential of the method, and its online algorithm also adapts to the training needs of large datasets, which has the advantage of strong real-time processing capability. As a combined prediction model based on deep neural networks, the TCN-CBAM-LSTM method offers significant advantages. The approach offers high accuracy and is capable of adapting to online training with large datasets, representing an innovative and effective soft sensor modeling strategy.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China(52467008), Key Project of Natural Science Foundation of Gansu Province(25JRRA150), Key Research and Development Planning Project of Gansu Province(23YFWA0007) , Lanzhou Science and Technology Plan Project (2023-1-16).

References

[1] Y. Jiang, S. Yin, J. Dong, and O. Kaynak, (2020) *"A review on soft sensors for monitoring, control, and optimization of industrial processes"* **IEEE Sensors Journal *21***(11): 12868–12881. DOI: 10.1109/JSEN.2020.3033153.

[2] J.-S. Wang and S. Han, (2015) *"Feed-Forward Neural Network Soft-Sensor Modeling of Flotation Process Based on Particle Swarm Optimization and Gravitational Search Algorithm"* **Computational Intelligence and Neuroscience *2015***(1): 147843. DOI: 10.1155/2015/147843.

[3] H. Kaneko and K. Funatsu, (2014) *"Application of online support vector regression for soft sensors"* **AIChE Journal *60***(2): 600–612. DOI: 10.1002/aic.14299.

[4] M. de Castro-Cros, S. Rosso, E. Bahilo, M. Velasco, and C. Angulo, (2021) *"Condition assessment of industrial gas turbine compressor using a drift soft sensor based in autoencoder"* **Sensors *21***(8): 2708. DOI: 10.3390/s21082708.

[5] M. Devakumar, G. Uma, M. Umapathy, et al., (2023) *"Critical measurement parameters estimation in liquid rocket engine using LSTM-based soft sensor"* **Flow Measurement and Instrumentation *92***: 102371. DOI: 10.1016/j.flowmeasinst.2023.102371.

[6] A. R. B. Abad, P. S. Tehrani, M. Naveshki, H. Ghorbani, N. Mohamadian, S. Davoodi, S. K.-y. Aghdam, J. Moghadasi, and H. Saberi, (2021) *"Predicting oil flow rate through orifice plate with robust machine learning algorithms"* **Flow Measurement and Instrumentation *81***: 102047. DOI: 10.1016/j.flowmeasinst.2021.102047.

[7] J. Fan, K. Zhang, Y. Huang, Y. Zhu, and B. Chen, (2023) *"Parallel spatio-temporal attention-based TCN for multivariate time series prediction"* **Neural Computing and Applications *35***(18): 13109–13118. DOI: 10.1007/s00521-021-05958-z.

[8] J. He, Y. Feng, and J. Zhu. "TCN Stock Price Prediction Model Based on Channel Attention Mechanism". In: *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*. IEEE. 2023, 850–855. DOI: 10.1109/ICCECT57938.2023.10140492.

[9] T. Limouni, R. Yaagoubi, K. Bouziane, K. Guissi, and E. H. Baali, (2023) *"Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model"* **Renewable Energy *205***: 1010–1024. DOI: 10.1016/j.renene.2023.01.118.

[10] H. V. Dudukcu, M. Taskiran, Z. G. C. Taskiran, and T. Yildirim, (2023) *"Temporal Convolutional Networks with RNN approach for chaotic time series prediction"* **Applied soft computing *133***: 109945. DOI: 10.1016/j.asoc.2022.109945.

[11] X. Yuan, S. Qi, Y. Wang, K. Wang, C. Yang, and L. Ye, (2021) *"Quality variable prediction for nonlinear dynamic industrial processes based on temporal convolutional networks"* **IEEE Sensors Journal *21***(18): 20493–20503. DOI: 10.1109/JSEN.2021.3096215.

[12] C. Sun, Y. Zhang, H. Zhao, H. Guo, Y. Zhang, and X. Hao, (2023) *"A soft sensor model for cement specific surface area based on TCN-ASRU neural network"* **IEEE Transactions on Instrumentation and Measurement *72***: 1–12. DOI: 10.1109/TIM.2023.3278292.

[13] W. Cheng, Y. Wang, Z. Peng, X. Ren, Y. Shuai, S. Zang, H. Liu, H. Cheng, and J. Wu, (2021) *"High-efficiency chaotic time series prediction based on time convolution neural network"* **Chaos, Solitons & Fractals *152***: 111304. DOI: 10.1016/j.chaos.2021.111304.

[14] T. H. T. Nguyen, N. Van Pham, V. N. N. Nguyen, H. M. Pham, Q. B. Phan, et al., (2022) *"Forecasting Wind Speed Using A Hybrid Model Of Convolutional Neural Network And Long-Short Term Memory With Boruta Algorithm-Based Feature Selection"* **Journal of Applied Science and Engineering *26***(8): 1053–1060. DOI: 10.6180/jase.202308_26(8).0001.

[15] Y. Ren, S. Wang, and B. Xia, (2023) *"Deep learning coupled model based on TCN-LSTM for particulate matter concentration prediction"* **Atmospheric Pollution Research *14***(4): 101703. DOI: 10.1016/j.apr.2023.101703.

[16] L. Fortuna, A. Rizzo, M. Sinatra, and M. G. Xibilia, (2003) *"Soft analyzers for a sulfur recovery unit"* **Control Engineering Practice *11***(12): 1491–1500. DOI: 10.1016/S0967-0661(03)00079-0.

[17] X. Yuan, C. Ou, Y. Wang, C. Yang, and W. Gui, (2020) *"Deep quality-related feature extraction for soft sensing modeling: A deep learning approach with hybrid VW-SAE"* **Neurocomputing *396***: 375–382. DOI: 10.1016/j.neucom.2018.11.107.

[18] X. Yuan, L. Li, Y. A. Shardt, Y. Wang, and C. Yang, (2020) *"Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development"* **IEEE Transactions on Industrial Electronics *68***(5): 4404–4414. DOI: 10.1109/TIE.2020.2984443.

[19] M. Mou and X. Zhao, (2022) *"Gated broad learning system based on deep cascaded for soft sensor modeling of industrial process"* **IEEE Transactions on Instrumentation and Measurement *71***: 1–11. DOI: 10.1109/TIM.2022.3170967.