ACS OMEGA

http://pubs.acs.org/journal/acsodf

Article

A New Regularized Spatiotemporal Attention-Based LSTM with Application to Nitrogen Oxides Emission Prediction

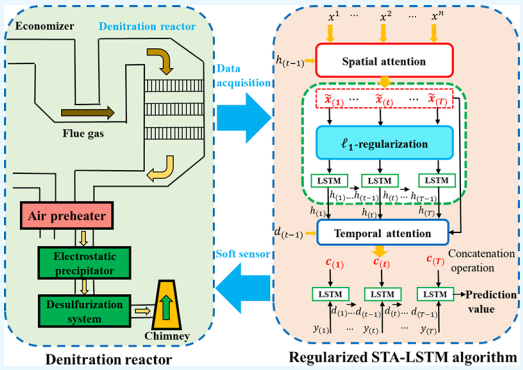
Xiuliang Wu, Kai Sun,* and Maoyong Cao*

Cite This: *ACS Omega* 2023, 8, 12853−12864

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: The data collected from complex process industries are usually time series with considerable nonlinearities and dynamics, as well as excessive redundancy. Moreover, there are temporal and spatial correlations between input variables and key performance variables. These characteristics bring great difficulties to data-driven modeling of the key performance variables. To overcome the problems, a new regularized spatiotemporal attention (STA)-based long short-term memory (LSTM) was developed. First, a standard LSTM network with an STA module was trained to capture the dynamic relationship between input and target variables. Second, the least absolute shrinkage and selection operator was introduced to optimize the STA module. Third, the hyperparameter representing the regularization strength of the algorithm was determined using a moving window cross-validation strategy. Finally, the proposed algorithm was compared to other state-of-the-art algorithms using artificial data, and then it was used to predict the nitrogen oxide emissions of a selective catalytic reduction denitration system. Simulation results showed that the proposed algorithm achieved more accurate predictions than the other algorithms. Furthermore, the statistics and analysis of the importance of the variables are consistent with known chemical-reaction mechanisms and observations of field experts. Thus, the proposed method can provide technical support for the predictive control and optimization of such systems.



1. INTRODUCTION

In modern industrial processes, there are key performance indicators that affect safety, efficiency, and product quality, which must be precisely monitored and controlled.[1] However, it is difficult to measure some indicators directly in real time owing to field conditions, technical constraints, and costs. Data-driven soft sensor models that infer hard-to-measure indicators from easy-to-measure indicators using specific algorithms for specific problems have the potential to solve these challenges.[2,3] The relationship between the indicators may be linear or nonlinear. Linear relationships are desirable because they reduce computation times and are easy to apply, and the models are easy to interpret. In recent years, linear methods, such as partial least squares,[4] principal component analysis,[5] and least absolute shrinkage and selection operator (LASSO),[6] have been applied extensively in data-driven modeling. However, if linear methods do not provide satisfactory results, nonlinear methods, such as neural networks (NNs),[7] support vector machines (SVM),[8] and Gaussian process regression[9] should be considered.

NNs are the most popular method of nonlinear modeling, and they provide powerful nonlinear mapping capabilities, efficient parallel computing, and excellent fault tolerance. Many NNs are available, including multilayer perceptron (MLP),[10] a stochastic configuration network,[11,12] a recurrent neural network (RNN),[13] and a long short-term memory (LSTM) network.[14] The LSTM network is an enhanced RNN that has three gated units to control the data flow. Information that is useful for long periods of time is stored well by the gate control, and the problems of gradient vanishing and explosion that occur with RNNs are alleviated.[15] Recently, LSTM-based modeling algorithms for time-series processes have become the subject of considerable interest. For example, Zhou et al. developed a novel data-driven modeling algorithm with different LSTMs for a grinding-classification process.[16] Shi et al. trained a data-driven model using state transition-LSTM and time-dimensional K-means to predict the quality of dual-sampling periods.[17] Xie et al. developed a variational autoencoder bidirectional LSTM and applied it to an actual grinding and classification process.[18] In addition, a sampling-interval-aware LSTM was developed to handle industrial data series with irregular sampling intervals and was applied to the soft sensor of a hydrocracking process.[19]

Received: December 26, 2022
Accepted: March 21, 2023
Published: March 30, 2023

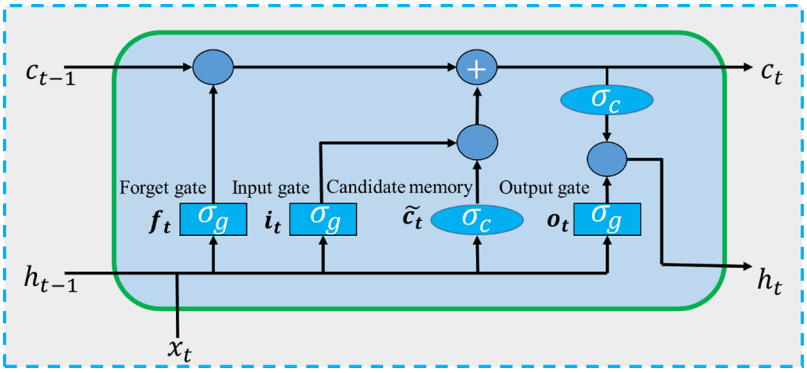


Figure 1. Schematic diagram showing the structure of a typical LSTM unit. The LSTM unit includes one memory cell and three gates. The terms are defined in Section 2.1.

LSTM is highly effective in handling long-term dependencies; however, it is difficult to capture the dynamic relationship between variables at different time steps.[20] To overcome this, Bahdanau et al. proposed an attention-based encoder-decoder to distinguish target-related hidden states, which has been widely studied and exploited.[21] Moreover, Feng et al. proposed a dual attention-based encoder-decoder method that utilized sequence-to-sequence learning.[22] Yu et al. developed a cascaded monitoring network algorithm to analyze the temporal and spatial information for the monitoring model.[23] In addition, a multihop attention graph convolutional network was proposed to capture the mutable characteristics of spatial coupling relations, and the effectiveness of the model was verified on a coal mill rig.[24] Yuan et al. developed a soft sensor using spatiotemporal attention-based LSTM (STA-LSTM) for industrial hydrocracking processes.[25] The attention mechanism allows the model to selectively capture important features and patterns while ignoring irrelevant or noisy information. Therefore, although the attention mechanism increases the model complexity, the performance of the deep learning networks is considerably improved.

However, large-scale industrial processes are usually very complex, with many input variables. The data collected from modern process industries are increasing rapidly due to the development and upgrading of automation technology. Excessive redundant variables increase the model complexity and lead to inaccurately estimated STA correlations between relevant variables and key quality variables.[26] Appropriate variable selection or regularization techniques can reduce model complexity and improve generalization performance.[27] For example, l_1-regularization, also called LASSO, is one of the most efficient regularization techniques for model reduction.[28] Sun et al. proposed an input selection approach for an MLP network with LASSO to predict the quality of kerosene produced by a crude distillation unit.[29] Ou et al.[30] proposed a quality-driven regularization for a stacked auto-encoder to capture quality-related variables from industrial process data and achieved desirable results. Moreover, Liu et al. used LASSO to remove redundant variables from relevant vector machine models.[31] In brief, these studies have demonstrated that the l_1 -regularization is very effective for model reduction and optimization for complex neural network models.

Selective catalytic reduction (SCR) technology is extensively applied to the denitration system of thermal power plants. In the system, the amount of nitrogen oxide (NO_x) emission is a key performance indicator related to environmental regulation and energy consumption. To facilitate the advanced control system and backup the hardware sensors, the development of a soft sensor for NO_x emission has been widely studied.[32] For example, to precisely estimate NO_x emission for optimizing the amount of injected ammonia, Li et al.[33] developed a soft sensor by combining the moving window partial least squares with locally weighted regression. Yang et al. proposed a dynamic model based on the least square SVM and mutual information to predict the NO_x concentration of a coal-fired boiler.[34] Furthermore, Wu et al. developed an effective soft sensor by combing the LASSO with the LSTM network to predict the NO_x emissions from a denitration system.[35]

To establish an accurate data-driven model of the NO_x emissions from a practical denitration system in a thermal power plant, a soft-sensing algorithm based on the LSTM network and regularized STA mechanism is proposed. The main contributions are summarized as follows:

(1) A new regularized STA-based LSTM (RSTA-LSTM) algorithm is developed, in which the LSTM is used for time series modeling, the STA is used to extract the dynamic correlations between input and target variables, and the l_1-regularization is utilized to simplify and optimize the STA.

(2) The proposed algorithm is compared with other state-of-the-art algorithms using an artificial dataset and an industrial dataset of an SCR denitration system. Comprehensive simulations and comparisons demonstrate the effectiveness of the proposed algorithm.

(3) The variable importance statistics and analysis are consistent with the chemical reactions and field experiences, which are conducive to the optimization and control of the system.

The rest of the work is arranged as follows: Section 2 reviews the preliminary theories related to the proposed algorithm. Section 3 describes the development of our approach in detail. Section 4 provides the simulation results of an artificial dataset. In Section 5, the developed algorithm is utilized to the data-driven modeling and statistical analysis of the practical industrial example. Finally, the concluding remarks and future works are provided in Section 6.

2. PRELIMINARY THEORY

2.1. LSTM. The structure of a typical LSTM unit, which includes one memory cell and three gates, is shown in Figure 1. At each time t, they are computed using the equations

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

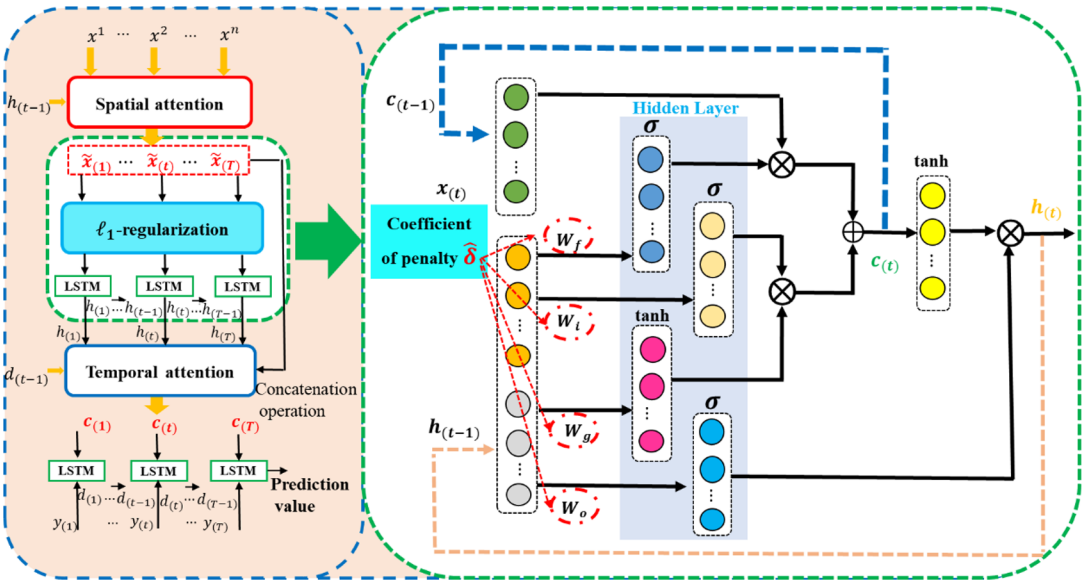


Figure 2. Schematic diagram showing the structure of the RSTA-LSTM algorithm. A regularization operator is added after the spatial attention calculation to reduce the complexity of the model.

$$f_{(t)} = \sigma_g(W_f[h_{(t-1)}; x_{(t)}] + b_f), \quad (1)$$

$$i_{(t)} = \sigma_g(W_i[h_{(t-1)}; x_{(t)}] + b_i), \quad (2)$$

$$o_{(t)} = \sigma_g(W_o[h_{(t-1)}; x_{(t)}] + b_o), \quad (3)$$

$$c_{(t)} = f_{(t)} \odot c_{(t-1)} + i_{(t)} \odot \sigma_c(W_c[h_{(t-1)}; x_{(t)}] + b_c), \text{ and} \quad (4)$$

$$h_{(t)} = o_{(t)} \odot \sigma_c(c_{(t)}). \quad (5)$$

Here, $f_{(t)}$, $i_{(t)}$, and $o_{(t)}$ represent the input, forget, and output gates, respectively; $c_{(t)}$ is the memory cell; and $h_{(t)}$ is the hidden state. Moreover, W_f, W_i, W_o, $W_c \in R^{m \times (m+n)}$ are weight matrices and b_f, b_i, b_o, $b_c \in R^m$ are bias parameters, where m denotes the number of hidden states and n denotes the number of input variables. The concatenation operation between $h_{(t-1)}$ and $x_{(t)}$ is $[h_{(t-1)}; x_{(t)}]$, σ_g is a sigmoid function, and σ_c is a hyperbolic tangent function. Finally, $\odot$ denotes the Hadamard product.

2.2. Spatiotemporal Joint Attention. To improve the performance of the model, spatiotemporal relationships were obtained by learning the dynamic relationships between variables and samples. For spatial attention, $x^k = (x^k_{(1)}, x^k_{(2)}, \cdots x^k_{(T)})^{tr} \in R^T$ is the kth input variable within a window of size T and the operator tr represents the transpose of the matrix. Thus, the attention weight is given by

$$e^i_{(t)} = v_e^{tr}\sigma_c(W_e[h_{(t-1)}; c_{(t-1)}] + U_e x^i_{(t)} + b_e), \quad (6)$$

and

$$\alpha^i_{(t)} = \sigma_z(e^i_{(t)}); i = 1, 2, \cdots, n. \quad (7)$$

Here, v_e, $b_e \in R^T$, $W_e \in R^{T \times 2m}$, and $U_e \in R^{T \times T}$ denote the parameters to be learned and $h_{(t-1)} \in R^m$ and $c_{(t-1)} \in R^m$ denote the hidden and cell states of the previous LSTM unit, respectively. At time t, the weights of the input variables are remapped by spatial attention to give

$$\tilde{x}_{(t)} = (\alpha^1_{(t)}x^1_{(t)}, \alpha^2_{(t)}x^2_{(t)}, \cdots, \alpha^n_{(t)}x^n_{(t)})^{tr}. \quad (8)$$

For temporal attention, the weighted input variables $x_{(\tilde{t})} \in R^{n \times T}$ and hidden state $h_j \in R^{m \times T}$ are combined. The relationships at time t are learned as

$$l^j_{(t)} = v_d^{tr}\sigma_c(W_d[d^0_{(t-1)}; c^0_{(t-1)}] + U_d[\tilde{x}_{(t)}; h_{(j)}] + b_d); j = 1, 2, \cdots, T, \quad (9)$$

and

$$\gamma^j_{(t)} = \sigma_z(l^j_{(t)}); j = 1, 2, \cdots, T, \quad (10)$$

where v_d, $b_d \in R^{m+n}$, $W_d \in R^{(m+n) \times 2q}$, and $U_d \in R^{(m+n) \times (m+n)}$ denote the parameters to be learned; q denotes the number of hidden states in the attention module; and $d^0_{(t-1)} \in R^q$ and $c^0_{(t-1)} \in R^q$ denote the hidden and cell states of the previous LSTM unit, respectively. The cell state is given by

$$\widetilde{h_{(t)}} = (\gamma^1_{(t)}\widetilde{h_{(1)}}, \gamma^2_{(t)}\widetilde{h_{(2)}}, \cdots, \gamma^T_{(t)}\widetilde{h_{(T)}})^{tr}, \quad (11)$$

and

$$\widetilde{c_{(t)}} = \sum_{j=1}^{T} \widetilde{h_{(t)}}. \quad (12)$$

The STA-LSTM algorithm recognizes significant input variables related to the output variable at different time steps and adaptively identifies the relationships between the hidden states and target variables.

2.3. LASSO. LASSO was used to alleviate overfitting by shrinking the estimated model coefficients. The ordinary least squares (OLS) loss was reformulated using l_1-regularization to penalize the absolute size of the weights. Thus, the model reduction was expressed as

$$L_{LASSO}(w) = L_{OLS}(\boldsymbol{w}), +\lambda |w|_1 \quad (13)$$

where $\boldsymbol{w} = [w_1, w_2, \cdots, w_n]^{tr}$ denotes the coefficient vector of the input variables and λ is a hyperparameter tuning the regularization strength between 0 and λ_{ub}. When $\lambda = 0$, LASSO is inactive. When λ_{ub} is sufficiently large, all the coefficients are forced to zero; hence, all the variables are

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

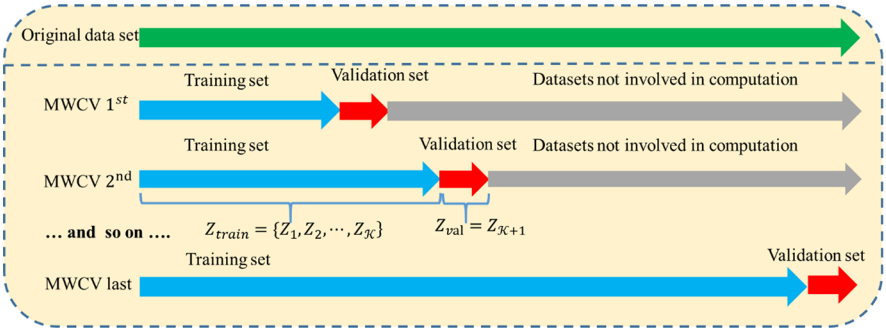


Figure 3. Schematic diagram summarizing the MWCV method. The MWCV method comprises two loops, the outer and inner loops, and it was used to determine the optimal λ.

removed. An appropriate value for λ is typically determined using cross-validation.

3. PROPOSED METHODOLOGIES

This section describes the development of the RSTA-LSTM algorithm for soft sensing of industrial multivariate dynamic processes.

3.1. Regularized STA Mechanism for LSTM. In the proposed algorithm, l_1-regularization is added to STA-LSTM as a penalty term to reduce model complexity. Figure 2 shows the structure of the proposed RSTA-LSTM algorithm in which a regularization operator $\delta = [\delta_1, \delta_2, \cdots, \delta_n]^{tr}$ is added to $\widetilde{x_{(t)}}$ after the spatial attention calculation. The hidden state then becomes

$$\widehat{h_{(t)}} = f_1(h_{(t-1)}, \delta \odot \widetilde{x_{(t)}}), \tag{14}$$

where f_1 is a single-layer LSTM with hidden size m.

The relationships for regularized STA can be expressed as

$$\widehat{h_{(t)}} = (\gamma_{(t)}^1 \widehat{h_{(1)}}, \gamma_{(t)}^2 \widehat{h_{(2)}}, \cdots, \gamma_{(t)}^T \widehat{h_{(T)}})^{tr}, \tag{15}$$

and

$$\widehat{c_{(t)}} = \sum_{j=1}^{T} \widetilde{h_{(t)}}. \tag{16}$$

To capture the relationships between the input and target variables, the target variable $y_{(t)}$ is concatenated to the feature vector $\widehat{c_{(t)}}$. That is,

$$\widetilde{y_{(t)}} = w^{tr}[y_{(t)}; \widehat{c_{(t)}}] + b, \tag{17}$$

where $w \in R^{m+1}$ and $b \in \mathrm{R}$ denote parameters to be learned.

Subsequently, the hidden state $d_{(t)}^0$ is updated to

$$\widetilde{d_{(t)}^0} = f_2(d_{(t-1)}^0, \widetilde{y_{(t-1)}}), \tag{18}$$

where f_2 is another single-layer LSTM with hidden size p.

Similarly, RSTA-LSTM performs the updating operation in eqs 17 and 18, which is given by

$$\widehat{y_{(t)}} = \tilde{w}^{tr}[y_{(t)}; \widehat{c_{(t)}}] + \tilde{b}, \tag{19}$$

and

$$\widehat{d_{(t)}^0} = f_3(\widetilde{d_{(t-1)}^0}, \widehat{y_{(t-1)}}), \tag{20}$$

where $\tilde{w} \in R^{m+1}$ and $\tilde{b} \in R$ denote the parameters to be learned and f_3 is a single-layer LSTM with hidden size q.

The multistep prediction using RSTA-LSTM is given by

$$\hat{Y} = v_y^{tr}\left(W_y\left[\widehat{d_{(t)}^0}; \widehat{c_{(t)}}\right] + b_{y1}\right) + b_{y2}, \tag{21}$$

where $\hat{Y} = (y_{(T+1)}, y_{(T+2)}, \cdots, y_{(T+\tau)})^{tr} \in R^{\tau}$ represents the prediction of the target variable in subsequent time steps τ. In addition, $W_y \in R^{q \times (q+m+n)}$ and $b_{y1} \in R^q$ map the concatenation $[d_{(t)}^0; c_{(t)}] \in R^{q+m+n}$ to the number of hidden decoder states. Predictions were obtained from the RSTA-LSTM algorithm using a linear function of the weight matrix $v_y \in R^{q \times \tau}$ and the bias vector $b_{y2} \in R^{\tau}$.

The optimization expression of RSTA-LSTM is given by

$$\hat{\delta} = \underset{\forall(X,Y)\in[\mathbb{X},\mathbb{Y}]}{\operatorname{argmin}} \sum \left\{ \sum (\mathrm{Y} - \hat{Y})^2 + \lambda |\delta|_1 \right\}, \tag{22}$$

where $|\delta|_1 = \sum_{i=1}^{n} |\delta_i|$ and Y are the true value. The nonlinear quadratic minimization problem in eq 22 can be solved using the active-set optimization algorithm reported by Hager and Zhang.[36] After that, the optimal $\hat{\delta}$ is added to the RSTA-LSTM to get the optimized LSTM network. For the active-set algorithm, it is easy to fall into local optima if the initial point starts from an inappropriate region. Therefore, the Monte Carlo method[37] is used to generate the initial solution, which is summarized as follows. First, a set of initial solution of δ are generated, in which each element is a random real number between 0 and 1. $\delta_i = 0$ means that the corresponding variable is deleted, while $\delta_i = 1$ means that the input weights of the corresponding variable remain unchanged. Second, each δ in the vector is added to the RSTA-LSTM to simulate the performance on the validation dataset. Finally, the δ with the best validation mean squared error (MSE) is taken as the initial solution of the active-set algorithm. To get a trade-off between algorithm performance and computational time, the number of random initial solutions is set to 200. Although this method cannot guarantee the global optimum, it can reduce the possibility of falling into local optima effectively.

In this study, the MSE was used as the loss function to train the models. The adaptive moment estimation (Adam) algorithm was adopted as the training method because it is superior to the root mean square backpropagation and momentum gradient descent algorithm.[38]

3.2. Choice of the Regularized Parameter. In the RSTA-LSTM, the regularized parameter λ controls the degree of shrinkage; hence, it has a critical effect on the performance of the model. In this study, an optimal value of λ was obtained through an enumerative search from parameter vector Λ, which contained predefined evenly distributed parameters between 0 and λ_{up}. In general, λ_{up} is a truncation value when the model performance deteriorates continuously as the number of input

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

variables decreases, while the number of the elements in the vector depends on the scale of the dataset.

Cross-validation is an effective method of evaluating the model's performance and selecting the hyperparameters. Considering the time series characteristics of industrial processes, a moving window cross-validation (MWCV) method[35] was used to determine the optimal λ. Briefly, this method comprises two loops. The outer loop enabled every parameter from vector $\Lambda = [0, \lambda_{up}]$ to be executed exactly once until all the parameters were traversed. In the inner loop, the MWCV was implemented as follows: (1) divide the raw dataset Z into K subsets, (2) train STA-LSTM with $Z_{train} = \{Z_1, Z_2, \cdots, Z_{\mathcal{K}}\}$, (3) calculate the new shrinkage coefficient $\hat{\delta}$, and (4) replace δ with $\hat{\delta}$ for the STA-LSTM and compute the MSE with $Z_{val} = Z_{\mathcal{K}+1}$. The stopping criterion for the MWCV method was that all subsets were traversed. Figure 3 shows a schematic of the MWCV method. Finally, the optimal value λ^* with the minimum $MSE_{mean}$was chosen from $[0, \lambda_{up}]$. The pseudocode for determining the regularized parameter is provided in Algorithm 1.

Algorithm 1 Pseudo-code for determining the parameter λ

Input: dataset $Z = \{\mathbb{X}, \mathbb{Y}\}$;
1. Initialization, $\lambda = 0$;
2. Divide Z into K subsets, $\{Z_1, Z_2, \cdots, Z_K\}$;
3. While $\lambda < \lambda_{up}$; //***Traverse the elements from the vector*** Λ
4. For $k = 1: K$
5. Train the STA-LSTM with $Z_{train} = \{Z_1, Z_2, \cdots, Z_{\mathcal{K}}\}$;
6. Introduce LASSO to the STA-LSTM, and get (14)-(21);
7. Obtain the new shrinkage coefficient $\hat{\delta}$ from (22);
8. Replace δ with $\hat{\delta}$ to get a new STA-LSTM model;
9. Validate the MSE with $Z_{test} = Z_{\mathcal{K}+1}$ for the current STA-LSTM;
10. End for
11. Average the MSE_{mean} of the current λ;
12. End while
13. Choose the optimal λ^* with minimum MSE_{mean} from $[0, \lambda_{up}]$;

Output: the optimal λ^*.

3.3. Computational Flow of the Overall Algorithm. The developed RSTA-LSTM is an iterative optimization over STA-LSTM with LASSO. A flowchart summarizing the process is shown in Figure 4. The algorithm begins by training an elementary STA-LSTM using the Adam optimization and backpropagation though time. Then, LASSO is introduced into the trained STA-LSTM to perform model reduction, and the appropriate regularized parameter λ is determined via the MWCV method. Subsequently, a new RSTA-LSTM model is obtained using the active-set optimization algorithm, and a new training dataset is obtained by the deletion of the input variables, which have $\hat{\delta} = 0$ from $\mathbb{X}$. The process is repeated until the termination conditions are met, when there is no improvement in the RSTA-LSTM model or the maximum number of iterations is reached, and the current model is considered to be the final model. The pseudocode of the proposed RSTA-LSTM is presented in Algorithm 2.

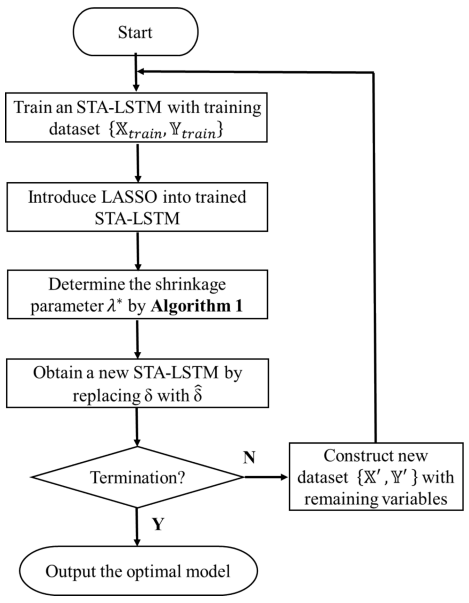


Figure 4. Flowchart showing the process of the proposed RSTA-LSTM. The RSTA-LSTM is an iterative optimization over STA-LSTM with LASSO.

Algorithm 2 Pseudo-code of the RSTA-LSTM

Input: dataset $Z = \{\mathbb{X}, \mathbb{Y}\}$;
Pre-tuned hyperparameters: learning rate η, training epoch $\mathcal{k}$, mini-batch size B, window size T, learning rate drop factor $\mathcal{f}$;
1. Standardize the dataset;
2. Divide the dataset into the training set $\mathbb{X}_{train}$, $\mathbb{Y}_{train}$, and testing set $\mathbb{X}_{test}$, $\mathbb{Y}_{test}$;
3. Set hyperparameters and find the maximum number of iterations ℓ for each epoch;
4. For i = 1 : $\mathcal{k}$
5. Update hidden state and cell state;
6. For j = 1 : ℓ
7. Train an LSTM with the training set;
8. Compute spatial attention from (6)-(8);
9. Compute temporal attention from (9)-(10);
10. Calculate the gradient by Adam optimization;
11. Calculate the MSE loss and update the hidden and cell states;
12. End for
13. Update learning rate at each epoch ($\eta = \eta \times \mathcal{f}$);
14. End for
15. While (termination conditions are not met);
//**see** ***the details of the conditional judgment in text***;
16. Add the operator $\delta = \{\delta_1, \delta_2, \cdots, \delta_n\}^{tr}$ to $\widetilde{x_{(t)}}$ and solve (14)-(21);
17. Choose the optimal λ^* from $[0, \lambda_{up}]$ by the **Algorithm 1**;
18. Calculate (22), and obtain $\hat{\delta}$;
19. Replace δ with $\hat{\delta}$ to generate a new STA-LSTM model;
20. Compute the MSE of the updated STA-LSTM with $\mathbb{X}_{test}$, $\mathbb{Y}_{test}$;
21. Replace the STA-LSTM model and the MSE if there are better ones;
22. Construct a new dataset $\{\mathbb{X}', \mathbb{Y}'\}$ with remaining variables;
23. End while

Output: The optimized model.

4. APPLICATION TO ARTIFICIAL DATASETS

4.1. Experimental Setting. The proposed RSTA-LSTM algorithm was tested using an artificial dataset with time-series characteristics. The split ratio of the dataset was 80:20, that is, 80% of the dataset was used for training and 20% was used for testing. Five popular algorithms, MLP, RNN, LSTM, STA-LSTM, and LASSO-LSTM, were used for comparison. All the algorithms were based on three types of elementary NN structures: MLP, RNN, and LSTM. The optimal hyperparameters of these NNs were selected using the grid search method. All the algorithms were programmed in MATLAB 2021a on a Windows 10.0 operating system, and they were executed on a machine learning server with an Intel(R) Core (TM) i7-10700 CPU and 64 GB of RAM.

The performance of each model was evaluated using the root mean squared error (RMSE), mean absolute percentage error (MAPE), and coefficient of determination (R^2), which were calculated using the equations

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_b - y_b')}, \tag{23}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{n=1}^{N}\frac{|y_b - y_b'|}{y_b}, \text{ and} \tag{24}$$

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

Table 1. Optimal Hyperparameters of Different Algorithms for an Artificial Dataset

hyperparameter	MLP	RNN	LSTM	STA-LSTM	LASSO-LSTM	RSTA-LSTM
input dimensions	50	50	50	50	50	50
output dimensions	1	1	1	1	1	1
length of input sequence		6	6	6	6	6
number of hidden neurons	10	10	50	50	50	50
number of fully connected layers	1	1	1	1	1	1
dropout rate			0.0002	0.0002	0.0002	0.0002
initial learning rate	0.001	0.001	0.003	0.003	0.003	0.003
mini batch size			10	10	10	10
max epochs	1000	300	300	300	300	300
λ					0.2	0.6

Table 2. Results of Different Algorithms on the Artificial Dataset

	R^2		RMSE		MAPE		
algorithm	mean	best	mean	best	mean	best	training time
MLP	0.1192	0.2035	6.7293	6.3480	0.1923	0.1810	0.79 s
RNN	0.1361	0.1935	7.3631	6.5071	0.2083	0.1877	40.12 s
LSTM	0.5995	0.6155	4.5062	4.4094	0.1249	0.1203	16.35 s
STA-LSTM	0.8083	0.8121	3.2912	3.2473	0.0961	0.0873	571.17 s
LASSO-LSTM	0.7909	0.8083	4.1446	3.8493	0.1166	0.1034	325.23 s
RSTA-LSTM	0.8256	0.8326	3.0511	2.9344	0.0801	0.0790	2721.25 s

$$R^2 = 1 - \frac{\sum_{n=1}^{N} (y_b - y_b')^2}{\sum_{n=1}^{N} (y_b - y)^2}, \quad (25)$$

where y_b is the actual value, y_b' is the predicted value, N is the total number of tested samples, and $\overline{y} = \text{mean}(y_b)$.

4.2. Friedman Dataset with Time Series. An artificial time-series dataset was designed based on the Friedman dataset,[39] which includes 50 input variables and 2000 samples. Each variable was uniformly distributed in the range [0, 1]. The response variable was generated using the equation

$$y_{(t)} = 10\sin(\pi x_{(t)}^1 x_{(t)}^2) + 20(x_{(t)}^3 - 0.5)^2 + 10x_{(t)}^4 + 5x_{(t)}^5 + \xi, \quad (26)$$

where

$$x_{(t)}^4 = 0.7x_{(t-1)}^4 + 0.9x_{(t-2)}^4 + 1.2x_{(t-3)}^4 + 0.7x_{(t-4)}^4 + 0.8x_{(t-5)}^4.$$

Here, ξ is white Gaussian noise and $x_{(t)}^4$ is designed to append a time series to the input data in which different coefficients represent different degrees of the time series.

The optimal hyperparameters of different algorithms are listed in Table 1. The statistical results, including the mean and best results of 10 runs, are listed in Table 2, which shows that RSTA-LSTM had better accuracy than the other algorithms. The combination of regularized methods and the STA mechanism improved the performance of a single optimization algorithm for LSTM while increasing the training time. In addition, LSTM was more suitable for an artificial dataset with time series of different durations than MLP or RNN.

To demonstrate the influence of λ on the algorithm performance, the prediction RMSE of the proposed algorithm with different λ is shown in Figure 5. In the case, the λ_{up} is set to 1.6 and there are nine elements evenly distributed in the vector $[0, \lambda_{up}]$. It can be seen from the figure that the smallest RMSE appears at $\lambda = 0.6$, meaning that the input variable coefficients are well shrunk and the model is properly optimized. When $\lambda = 0$, there is no any shrinkage on the input variables, and the input redundancy makes the model performance inadequate. As the λ increases, more and more relevant variables are removed, which makes the model performance continue to deteriorate.

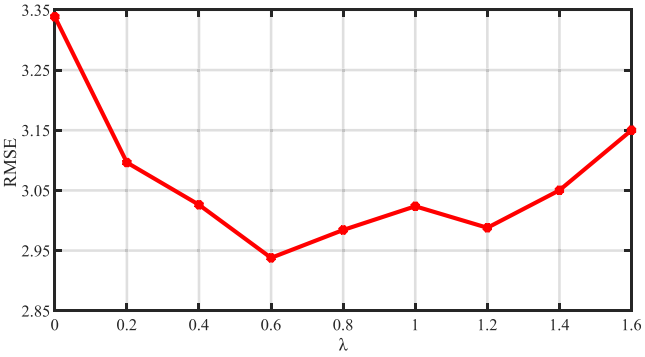


Figure 5. Prediction RMSE of different λ for the artificial dataset. The algorithm has the smallest RMSE when $\lambda = 0.6$.

The fitting results of the proposed algorithm are presented in Figure 6, which shows that the proposed algorithm accurately predicted the target variable. Figure 7 shows the distributions of the errors between the real and predicted values for the different algorithms. Among all the algorithms, the box plot for RSTA-LSTM is the most compact and its median is the closest to zero. Therefore, the proposed algorithm has superior accuracy for data-driven modeling of an artificial dataset.

5. APPLICATION TO A DENITRIFICATION PROCESS

5.1. Description and Analysis of the Process. SCR denitration systems are commonly applied in power plant boilers and other combustion scenarios to remove NO_x from flue gas emissions. A brief flowchart representing the SCR denitration system used in this study is shown in Figure 8. The SCR system was located between the economizer and air preheater in the thermal power plant. Flue gas from the

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

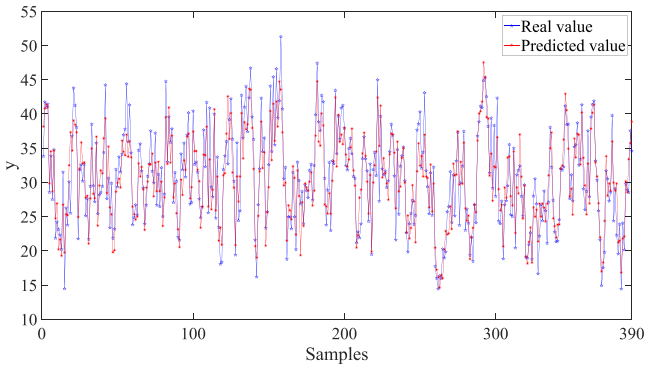


Figure 6. Comparison of real and predicted results. Real and predicted values obtained using RSTA-LSTM.

economizer passed through the ammonia spray grid of the denitrification reactor, wherein a mixture of gaseous ammonia and air was injected into the flue gas. Then, the mixture was passed through the catalyst layer and NO_x was converted to N_2 and H_2O. The chemical reactions were

$$4NO + 4NH_3 + O_2 \rightarrow 4N_2 + 6H_2O \quad (27)$$

and

$$6NO_2 + 8NH_3 \rightarrow 7N_2 + 12H_2O. \quad (28)$$

After these reactions, most of the NO_x is removed from the flue gas. Next, the flue gas flows toward the air preheater and the electrostatic precipitator removes any dust. Finally, the flue gas enters the desulfurization system, which removes SO_2, and it is discharged into the atmosphere through the chimney.

According to Chinese national ambient air quality standards, NO_x emissions are limited to 50 mg/Nm^3 when the O_2 concentration at the exit is 6%. An online analyzer was installed approximately 5 m from the chimney inlet. A 50 m sampling pipe was used to connect the analyzer to the continuous emission monitoring system. After long-term operation, dust accumulated in the pipe and blocked the flow of gas. Therefore, regular back blowing and calibrations were required to ensure the stability of the analyzer. During this period, the analyzer could not operate normally owing to the sampling interruptions. According to field statistics, the analyzers on the four denitration units at the plant malfunction several times per year. Thus, a data-driven soft sensor model of NO_x emissions is a valuable technique that could guarantee continuous results.

In the SCR process, ammonia injection is controlled to reduce ammonia slip and ensure NO_x reduction. However, the length of the pipeline means that the sampling time delay of the measurement system cannot be determined precisely; hence, it is difficult for the controller to accurately regulate ammonia injection. To guarantee that the NO_x emissions are within national limits, the field controller usually oversprays ammonia to increase NO_x absorption. However, excessive spraying increases ammonia slip, which wastes energy and causes additional pollution.

In summary, there are obvious dynamics and time delays of unknown durations in the SCR process. Therefore, to achieve accurate monitoring and optimal process control, it is necessary to develop a precise data-driven soft sensor. It has been shown that the proposed algorithm is an effective method of modeling dynamic processes with time delays of various durations; therefore, it was applied to the development of a soft sensor for the SCR process.

5.2. Simulation Results and Comparisons. A dataset including 3980 samples was obtained from the denitration system of a thermal power plant in East China. There were a total of 45 input variables, which included 15 variables from the boiler and generator, 7 variables from the draft fan, and 23 variables from the reactor and flue gas. The output variable was the concentration of NO_x in the flue gas at the chimney outlet. All the variables are listed in Table 3.

After some trials, the optimal hyperparameters of different algorithms are given in Table 4. The statistical performance of the different algorithms after 10 cycles is shown in Table 5. The LSTM-based models performed significantly better than the MLP and RNN models. This is because the SCR process had explicit dynamics and a large time delay owing to the long sampling pipeline and complex reaction mechanism, whereas the LSTM networks had inherent advantages when modeling processes with long-term time series. Moreover, compared with STA-LSTM and LASSO-LSTM, the proposed algorithm achieved better results for all performance indexes. This shows that -regularization improved the capability of the STA to capture the complex dynamic correlation between the input and output variables.

Although our approach has the highest training time, its accuracy is significantly better than other algorithms. Actually,

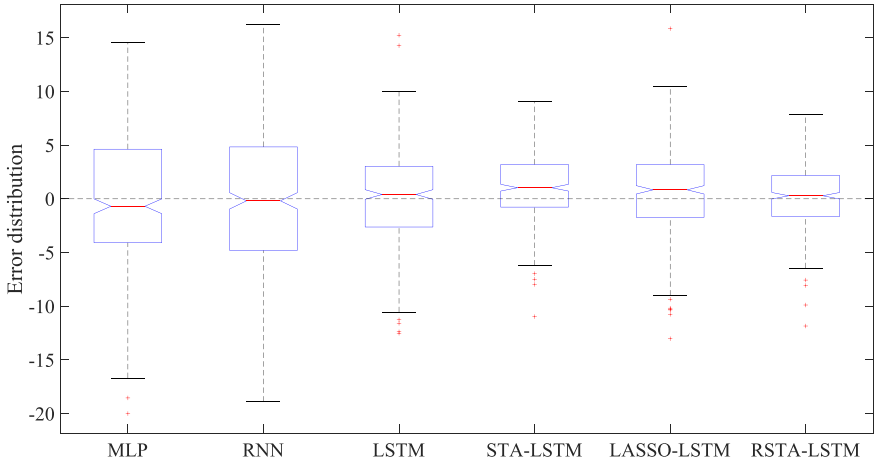


Figure 7. Prediction errors for different algorithms. Distributions of the errors between the real and predicted values for different algorithms.

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

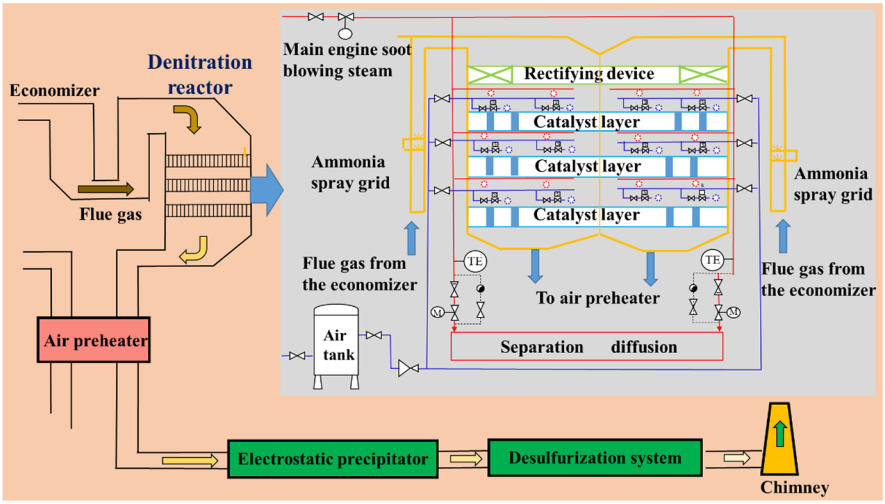


Figure 8. Flow chart representing the SCR denitration system. The SCR system was located between the economizer and air preheater.

Table 3. Input and Output Variables of the SCR Process[a]

type	no.	name	unit	no.	name	unit
process variables	1	NO_2 conc at the left inlet	mg/Nm^3	24	6 kV SC of induced draft fan A	A
	2	NO_2 conc at the right inlet	mg/Nm^3	25	6 kV SC of induced draft fan B	A
	3	ammonia flow at mixer inlet A	Nm^3/h	26	6 kV SC of forced draft fan A	A
	4	ammonia flow at mixer inlet B	Nm^3/h	27	6 kV SC of forced draft fan B	A
	5	converted NO_x conc at the left outlet	mg/Nm^3	28	SC of primary fan in frequency conversion cabinet A	A
	6	converted NO_x conc at the right outlet	mg/Nm^3	29	SC of primary fan in frequency conversion cabinet B	A
	7	total air volume into the boiler	ton/h	30	FGT at the outlet of the final superheater side A	°C
	8	main steam flow rate	ton/h	31	FGT at the outlet of the final superheater side B	°C
	9	converted O_2 conc	%	32	FGT at the left inlet	°C
	10	total coal into the boiler	ton/h	33	FGT at the right inlet	°C
	11	active power of the generator	kW	34	O_2 conc at the left inlet	%
	12	main steam press.	Pa	35	O_2 conc at the left outlet	%
	13	main steam temp.	°C	36	O_2 conc at the right inlet	%
	14	reheater press.	Pa	37	O_2 conc at the right outlet	%
	15	reheater temp. at the outlet	°C	38	ammonia escape rate at the left	PPM
	16	degree of superheat	°C	39	ammonia escape rate at the right	PPM
	17	total water into the boiler	ton/h	40	CO conc at the left inlet	%
	18	furnace press.	Pa	41	CO conc at the right inlet	%
	19	primary air press.	Pa	42	valve opening at mixer inlet A	%
	20	exhaust gas temp.	°C	43	valve opening at mixer inlet B	%
	21	total spray water flow of superheater	ton/h	44	coal quantity of feeder B	ton
	22	DPOFGI and O at air pre-heater A	Pa	45	coal quantity of feeder E	ton
	23	DPOFGI and O at air pre-heater B	Pa			
output	1	NO_x emissions of the denitration system				mg/Nm^3

[a]DPOFGI and O, differential pressure of flue gas at inlet and outlet; FGT, flue gas temperature; SC, side current.

Table 4. Optimal Hyperparameters of Different Algorithms for the SCR Process

hyperparameter	MLP	RNN	LSTM	STA-LSTM	LASSO-LSTM	RSTA-LSTM
input dimensions	45	45	45	45	45	45
output dimensions	1	1	1	1	1	1
length of input sequence		5	5	5	5	5
number of hidden neurons	10	10	50	50	50	50
number of fully connected layers	1	1	1	1	1	1
dropout rate			0.0002	0.0002	0.0002	0.0002
initial learning rate	0.001	0.001	0.002	0.002	0.002	0.002
mini batch size			5	5	5	5
max epochs	1000	300	300	300	300	300
λ					0.1	0.3

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

Table 5. Results of Different Algorithms on the NO_x Emissions Prediction

	R^2		RMSE		MAPE		
algorithm	mean	best	mean	best	mean	best	training time
MLP	0.5767	0.6265	4.6451	3.9381	0.1122	0.0890	5.25 s
RNN	0.4935	0.6605	5.1128	3.8832	0.1186	0.0925	52.35 s
LSTM	0.6630	0.6839	3.9363	3.6582	0.0939	0.0912	31.27 s
STA-LSTM	0.7324	0.7545	3.7812	3.6189	0.0961	0.0855	1474.52 s
LASSO-LSTM	0.7325	0.7883	3.7587	3.3997	0.0931	0.0863	565.63 s
RSTA-LSTM	0.7912	0.8069	3.3682	3.1229	0.0701	0.0649	6102.32 s

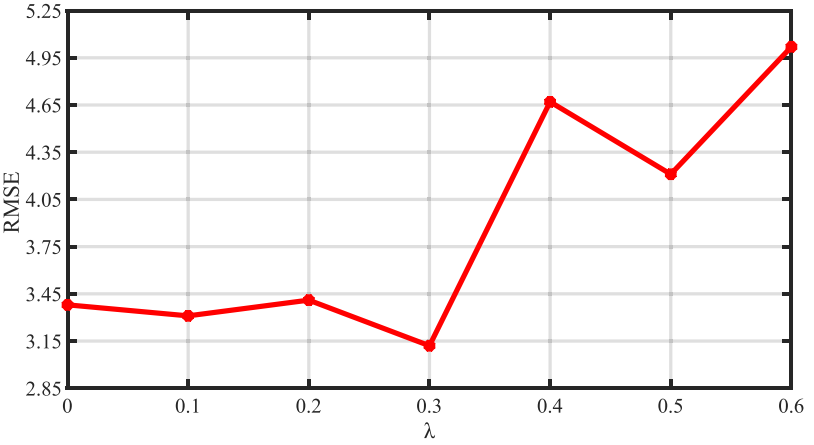


Figure 9. Prediction RMSE of different λ for the SCR process. The smallest RMSE appears at $\lambda = 0.3$.

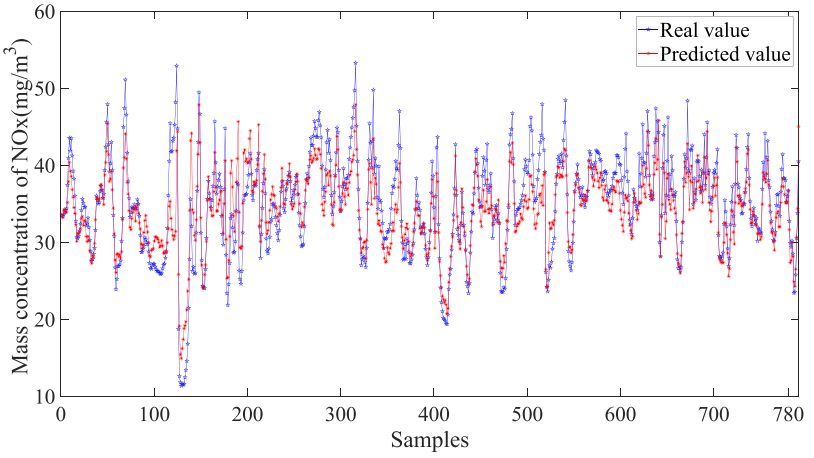


Figure 10. Comparison of measured and predicted NO_x emissions. Measured and predicted values obtained using RSTA-LSTM.

the training time of the soft sensors is not the crucial factor for the advanced control system. The models are often trained offline and then are duplicated and executed as subroutines in the industrial computer. Despite their different training time, the predicting time of these soft sensors is very close, which can totally meet the real-time requirements of online predictions.

For the dataset of the SCR process, the λ_{up} is set to 0.6 and there are seven elements in the vector $[0, \lambda_{up}]$. The prediction RMSE of the proposed algorithm with different λ for the SCR process is given in Figure 9. The algorithm has the smallest RMSE when $\lambda = 0.3$.

The measured NO_x emissions and those predicted by the RSTA-LSTM algorithm are shown in Figure 10. This demonstrates that the proposed algorithm can produce very accurate predictions.

The distributions of the errors between the real and measured values for the different algorithms are shown in Figure 11. The proposed algorithm had the smallest error range and the flattest box than the existing algorithms. In summary, after comprehensive comparisons, the proposed algorithm was superior to the existing algorithms in the soft sensor modeling of NO_x emissions from the SCR process.

5.3. Statistical Analysis and Discussion on Variable Importance. To verify the effectiveness of the proposed RSTA mechanism, an indicator called the permutation variable importance (PVI)[40] was used to measure the importance of each input variable to the output variable. The PVI of an input variable is obtained by calculating the ratio between the prediction errors before and after the variable data are shuffled; only the data corresponding to the variable are shuffled each time. The process of computing the PVI of variable v is as

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

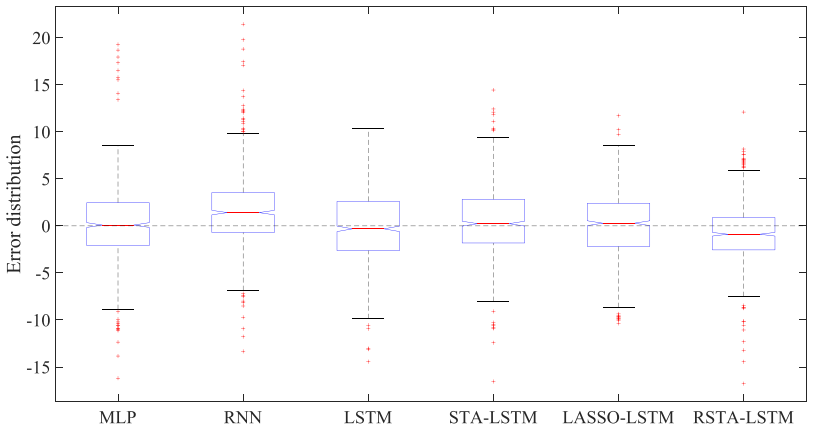


Figure 11. Prediction errors for different algorithms. Distributions of the errors between the measured and predicted values for different algorithms.

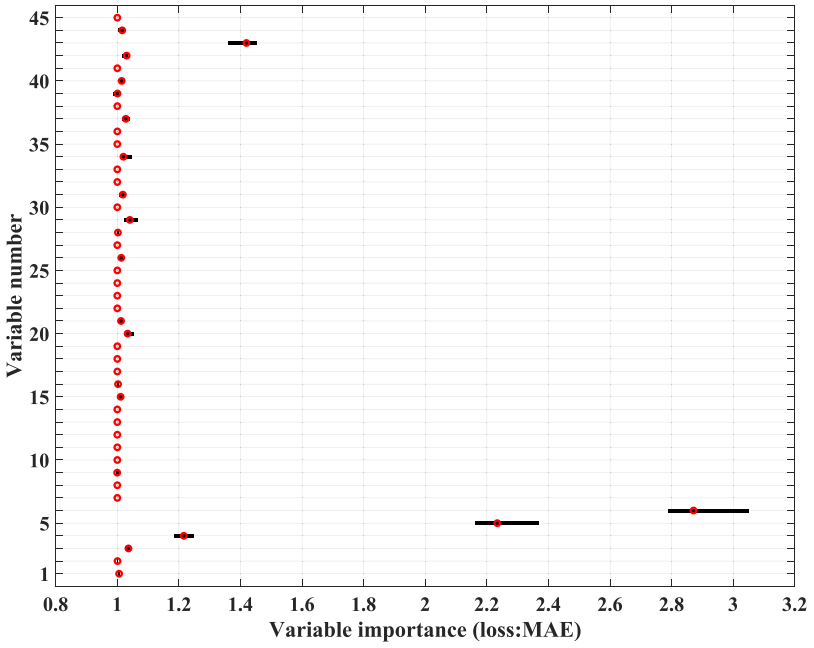


Figure 12. Importance of the candidate input variables with MAE. The distributions of the PVIs are represented by black lines and the median PVIs are denoted by red circles.

follows: (1) the original mean absolute error (MAE) of the original dataset $\{\mathbb{X}_{test}, \mathbb{Y}_{test}\}$ is calculated, (2) the data corresponding to v in the testing set $\mathbb{X}_{test}$ are randomly shuffled, (3) the updated MAE of the shuffled dataset $\{\widetilde{\mathbb{X}_{test}}, \mathbb{Y}_{test}\}$ with the trained model is calculated, and (4) the ratio between the updated and original MAE is calculated. This process was repeated for each variable, and their PVIs were obtained.

A larger PVI indicates that the variable is more important. If the PVI of a variable is very close to one, then the variable is irrelevant to the output variable. In this study, the shuffling process was repeated 10 times to obtain statistical results from which the 5%, median, and 95% quantiles of the PVI were calculated. In Figure 12, the distributions of the PVIs are represented by black lines and the median PVIs are denoted by red circles.

Four variables significantly affected the output variable. The most significant variable was variable 6, that is, the converted NO_x concentration at the right outlet. This variable describes the NO_x concentration upstream; hence, it is highly relevant to the output variable. Similarly, the second most significant variable was variable 5, that is, the converted NO_x concentration at the left outlet.

In the early stage of the SCR process, liquid ammonia is preheated to produce gaseous ammonia, which is mixed with air from the dilution fan in the mixer. The NO_x in the flue gas undergoes a denitrification reaction with gaseous ammonia under the action of the catalyst, as shown in eqs 27 and 28. Two flow valves control the amount of ammonia available for the denitrification reaction. In practice, the valve at inlet A is usually kept at a fixed opening, whereas the valve at inlet B is adjusted to regulate the flow of ammonia. Consequently, variable 43, which is the control valve opening at mixer inlet B, and variable 4, which is the ammonia flow at mixer inlet B, have a significant effect on the output variable, as shown in Figure 12.

In summary, the PVI statistics are consistent with the process mechanism and expert experience, which verifies the interpretability of the model using the proposed RSTA-LSTM. The developed soft sensor and feature importance analysis provide a solid foundation for the optimization and control of this process.

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

6. CONCLUSIONS

In this study, a soft-sensor algorithm that combines LASSO with STA-LSTM was proposed for highly complex and dynamic processes. The proposed algorithm introduces l_1-regularization into the STA-LSTM and performs shrinkage on the input weights of the STA-LSTM for model reduction. This approach combines the ability of STA to capture dynamic relationships with the superior weight shrinkage of l_1-regularization. As an industrial application, the proposed algorithm was applied to an artificial dataset and used to predict NO_x emissions of an SCR process at a thermal power plant. The simulations and statistical analysis show that the proposed algorithm has advantages over other advanced algorithms in terms of accuracy and interpretability. The developed soft sensor provides accurate and reasonable data-driven models that can be used to upgrade SCR control systems.

However, the data-driven model with the proposed algorithm was trained offline using historical measured data. Therefore, if the field conditions change or there are concept drifts in some key variables, the performance may deteriorate. Future research will focus on online learning and updating of the strategies used by the algorithm.

AUTHOR INFORMATION

Corresponding Authors

Kai Sun − *School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China; State Key Laboratory of Process Automation in Mining and Metallurgy, Beijing 100160, China; Beijing Key Laboratory of Process Automation in Mining and Metallurgy, Beijing 100160, China;* orcid.org/0000-0002-9482-470X; Email: sunkai79@qlu.edu.cn

Maoyong Cao − *College of Electrical Engineering and Automation, Shandong University of Science and Technology (SDUST), Qingdao 266590, China; School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China;* Email: cmy@qlu.edu.cn

Author

Xiuliang Wu − *College of Electrical Engineering and Automation, Shandong University of Science and Technology (SDUST), Qingdao 266590, China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.2c08205

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The work is partially supported by the Open Foundation of State Key Laboratory of Process Automation in Mining and Metallurgy (BGRIMM-KZSKL-2021-07), the Shandong Provincial Natural Science Foundation of China (ZR2021MF022), and the National Major Science and Technology Projects of China (2020YFB1711200).

REFERENCES

(1) Sun, Q.; Ge, Z. A survey on deep learning for data-driven soft sensors. *IEEE Transactions on Industrial Informatics* **2021**, *17*, 5853−5866.

(2) Liu, Y.; Xie, M. Rebooting data-driven soft-sensors in process industries: A review of kernel methods. *J. Process Control* **2020**, *89*, 58−73.

(3) Zheng, J.; Ma, L.; Wu, Y.; Ye, L.; Shen, F. Nonlinear dynamic soft sensor development with a supervised hybrid CNN-LSTM network for industrial processes. *ACS Omega* **2022**, 16653.

(4) Jiang, Q.; Yan, X.; Yi, H.; Gao, F. Data-driven batch-end quality modeling and monitoring based on optimized sparse partial least squares. *IEEE Trans. Ind. Electron.* **2019**, *67*, 4098−4107.

(5) Kazemi, P.; Bengoa, C.; Steyer, J.-P.; Giralt, J. Data-driven techniques for fault detection in anaerobic digestion process. *Process Saf. Environ. Prot.* **2021**, *146*, 905−915.

(6) Yazdi, M.; Golilarz, N. A.; Nedjati, A.; Adesina, K. A. An improved lasso regression model for evaluating the efficiency of intervention actions in a system reliability analysis. *Neur. Comput. Appl.* **2021**, *33*, 7913−7928.

(7) Sayahi, T.; Garff, A.; Quah, T.; Le, K.; Becnel, T.; Powell, K. M.; Gaillardon, P. E.; Butterfield, A. E.; Kelly, K. E. Long-term calibration models to estimate ozone concentrations with a metal oxide sensor. *Environ. Pollut.* **2020**, 267.

(8) Abdelaal, A.; Elkatatny, S.; Abdulraheem, A. Data-driven modeling approach for pore pressure gradient prediction while drilling from drilling parameters. *ACS Omega* **2021**, *6*, 13807−13816.

(9) Kanno, Y.; Kaneko, H. Ensemble just-in-time model based on Gaussian process dynamical models for nonlinear and dynamic processes. *Chemom. Intell. Lab. Syst.* **2020**, *203*, No. 104061.

(10) Lavrentev, F. V.; Rumyantsev, I. S.; Ivanov, A. S.; Shilovskikh, V. V.; Orlova, O. Y.; Nikolaev, K. G.; Andreeva, D. V.; Skorb, E. V. Soft hydrogel actuator for fast machine-learning-assisted bacteria detection. *ACS Appl. Mater. Interfaces* **2022**, *14*, 7321−7328.

(11) Dai, W.; Zhou, X.; Li, D.; Zhu, S.; Wang, X. Hybrid parallel stochastic configuration networks for industrial data analytics. *IEEE Transactions on Industrial Informatics* **2021**, *18*, 2331−2341.

(12) Dai, W.; Li, D.; Zhou, P.; Chai, T. Stochastic configuration networks with block increments for data modeling in process industries. *Inf. Sci.* **2019**, *484*, 367−386.

(13) Geng, J.; Yang, C.; Li, Y.; Lan, L.; Luo, Q. MPA-RNN: a novel attention-based recurrent neural networks for total nitrogen prediction. *IEEE Transactions on Industrial Informatics* **2022**, 6516.

(14) Han, Y.; Ding, N.; Geng, Z.; Wang, Z.; Chu, C. An optimized long short-term memory network based fault diagnosis model for chemical processes. *J. Process Control* **2020**, *92*, 161−168.

(15) Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235−1270.

(16) Zhou, J.; Wang, X.; Yang, C.; Xiong, W. A novel soft sensor modeling approach based on Difference-LSTM for complex industrial process. *IEEE Transactions on Industrial Informatics* **2022**, *18*, 2955−2964.

(17) Shi, X.; Li, Y.; Yang, Y.; Sun, B.; Qi, F. Multi-models and dual-sampling periods quality prediction with time-dimensional K-means and state transition-LSTM network. *Inf. Sci.* **2021**, *580*, 917−933.

(18) Xie, W.; Wang, J.; Xing, C.; Guo, S.; Guo, M.; Zhu, L. Variational autoencoder bidirectional long and short-term memory neural network soft-sensor model based on batch training strategy. *IEEE Transactions on Industrial Informatics* **2021**, *17*, 5325−5334.

(19) Yuan, X.; Li, L.; Wang, K.; Wang, Y. Sampling-interval-aware LSTM for industrial process soft sensing of dynamic time sequences with irregular sampling measurements. *IEEE Sens. J.* **2021**, *21*, 10787−10795.

(20) Liu, Y.; Zhang, Q.; Song, L.; Chen, Y. Attention-based recurrent neural networks for accurate short-term and long-term dissolved oxygen prediction. *Comp. Electron. Agric.* **2019**, *165*, No. 104964.

(21) Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* 2014, DOI: 10.48550/arXiv.1409.0473.

(22) Feng, L.; Zhao, C.; Sun, Y. Dual attention-based encoder−decoder: A customized sequence-to-sequence learning for soft sensor

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864

development. *IEEE Transactions on Neural Networks and Learning Systems* **2021**, *32*, 3306−3317.
(23) Yu, W.; Zhao, C.; Huang, B. MoniNet with concurrent analytics of temporal and spatial information for fault detection in industrial processes. *IEEE Transactions on Cybernetics* **2022**, *52*, 8340−8351.
(24) Zhu, K.; Zhao, C. Dynamic graph-based adaptive learning for online industrial soft sensor with mutable spatial coupling relations. *IEEE Trans. Ind. Electron.* **2022**, 1.
(25) Yuan, X.; Li, L.; Shardt, Y. A.; Wang, Y.; Yang, C. Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development. *IEEE Trans. Ind. Electron.* **2020**, *68*, 4404−4414.
(26) Qma, B.; Shuai, T. A.; Jia, W. A.; Jw, A.; Wwyna, B. Attention-based spatio-temporal dependence learning network. *Inf. Sci.* **2019**, *503*, 92−108.
(27) Long, J.; Li, T.; Yang, M.; Hu, G.; Zhong, W. Hybrid strategy integrating variable selection and a neural network for fluid catalytic cracking modeling. *Ind. Eng. Chem. Res.* **2018**, *58*, 247−258.
(28) Fan, Y.; Tao, B.; Zheng, Y.; Jang, S. S. A data-driven soft sensor based on multilayer perceptron neural network with a double LASSO approach. *IEEE Transactions on Instrumentation and Measurement* **2020**, *7*, 69.
(29) Sun, K.; Huang, S.-H.; Wong, D. S.-H.; Jang, S.-S. Design and application of a variable selection method for multilayer perceptron neural network with LASSO. *IEEE transactions on neural networks and learning systems* **2017**, *28*, 1386−1396.
(30) Ou, C.; Zhu, H.; Shardt, Y. A.; Ye, L.; Yuan, X.; Wang, Y.; Yang, C. Quality-driven regularization for deep learning networks and its application to industrial soft sensors. *IEEE Transactions on Neural Networks and Learning Systems* **2022**, 1.
(31) Liu, Y.; Liu, B.; Zhao, X.; Xie, M. Development of RVM-based multiple-output soft sensors with serial and parallel stacking strategies. *IEEE Transactions on Control Systems Technology* **2019**, *27*, 2727−2734.
(32) Xie, P.; Gao, M.; Zhang, H.; Niu, Y.; Wang, X. Dynamic modeling for NOx emission sequence prediction of SCR system outlet based on sequence to sequence long short-term memory network. *Energy* **2020**, *190*, 116482.
(33) Li, Z.; Lee, Y.-S.; Chen, J.; Qian, Y. Developing variable moving window PLS models: using case of NOx emission prediction of coal-fired power plants. *Fuel* **2021**, *296*, No. 120441.
(34) Yang, T.; Ma, K.; Lv, Y.; Bai, Y. Real-time dynamic prediction model of NOx emission of coal-fired boilers under variable load conditions. *Fuel* **2020**, *274*, No. 117811.
(35) Wu, X.; Yu, X.; Xu, R.; Cao, M.; Sun, K. Nonlinear dynamic soft-sensing modeling of NOx emission of a selective catalytic reduction denitration system. *IEEE Transactions on Instrumentation and Measurement* **2022**, *71*, 1−11.
(36) Hager, W. W.; Zhang, H. A new active set algorithm for box constrained optimization. *SIAM Journal on Optimization* **2006**, *17*, 526−557.
(37) Metropolis, N.; Ulam, S. The monte carlo method. *J. Am. Stat. Assoc.* **1949**, *44*, 335−341.
(38) Da, K. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* 2014, DOI: 10.48550/arXiv.1412.6980.
(39) Friedman, J. H. Multivariate adaptive regression splines. *Annal Stat.* **1991**, *19*, 1−67.
(40) Fisher, A.; Rudin, C.; Dominici, F. Model class reliance: Variable importance measures for any machine learning model class, from the "Rashomon" perspective. *arXiv preprint arXiv:1801.01489* 2018, *68*.

https://doi.org/10.1021/acsomega.2c08205
ACS Omega 2023, 8, 12853−12864