Research Article

Development of a Novel Soft Sensor with Long Short-Term Memory Network and Normalized Mutual Information Feature Selection

Dongfeng Li,1 Zhirui Li,2 and Kai Sun3

1Shandong Experimental High School, Jinan 250001, China
2School of Microelectronics, Shandong University, Jinan 250001, China
3School of Electrical Engineering and Automation, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China

Correspondence should be addressed to Kai Sun; sunkai79@qlu.edu.cn

Received 1 February 2020; Accepted 21 February 2020; Published 25 April 2020

Guest Editor: Ping Zhao

Copyright © 2020 Dongfeng Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a novel soft sensor is developed by combining long short-term memory (LSTM) network with normalized mutual information feature selection (NMIFS). In the proposed algorithm, LSTM is designed to handle time series with high nonlinearity and dynamics of industrial processes. NMIFS is conducted to perform the input variable selection for LSTM to simplify the excessive complexity of the model. The developed soft sensor combines the excellent dynamic modelling of LSTM and precise variable selection of NMIFS. Simulations on two actual production datasets are used to demonstrate the performance of the proposed algorithm. The developed soft sensor could precisely predict the objective variables and has better performance than other methods.

1. Introduction

Due to technological constraints, sensor characteristics, environmental factors, etc., many variables cannot be measured or the measurement frequency is very low in actual industrial processes. Soft measurement provides an excellent solution to construct mathematical models from easily measured variables to hard ones [1–3]. Neural networks (NNs) are advanced methods that can precisely model complex and nonlinear system and therefore have been widely used in soft sensors [4–6]. Heidari et al. [7] developed a new multi-layer perceptron (MLP) network to estimate nanofluid relative viscosity, which are more accurate than other NN structures. Sheela and Deepa [8] designed a synthesized model by combining self-organizing maps (SOMs) with MLP and then applied it to forecast the wind speed of a renewable energy process. He et al. [9] developed an auto-associative hierarchical NN for a soft sensor of chemical processes, and its application to a purified terephthalic acid solvent process demonstrated the effectiveness of the algorithm. Zabadaj et al. [10] proposed an effective soft sensor for the supervisory control of biotransformation production, and the efficiency of the approach was demonstrated. Rehrla et al. [11] developed a soft sensor method for estimating the active pharmaceutical ingredient concentration from the system data and the soft sensor model was tested in the three different continuous production lines. A novel approach of supervised latent factor analysis was proposed based on system data regression modelling, which can effectively predict heterogeneous variances, and soft sensors are established for quality estimation in the two case studies [12].

However, industrial systems are intrinsic complex and have high temporal correlations between dataset samples. That is, process data are time series with strong nonlinearities and dynamics, which increases the difficulty of modelling with conventional NNs. Recently, a powerful type of NN named long short-term memory (LSTM) was...
designed to handle sequence dependence [13–15]. An LSTM network is more significant in learning long-term temporal dependencies since its memory cells can maintain its state over a long time and standardize the information moving into and out of the cell. Therefore, LSTM networks have been effectively used in many different fields, such as precipitation nowcasting [16], traffic forecasting [17], human action recognition [18], etc. Due to its advantages, LSTM has also been applied in soft sensor development in industrial processes. Yuan et al. developed a supervised LSTM network for a soft sensor and demonstrated the superiority of the proposed soft sensor by two actual industrial datasets [19]. Sun proposed a new LSTM network by combining unsupervised feature selection and supervised dynamic modelling methods for a soft sensor and validated the network by a practical CO₂ absorption column [20].

The rapid evolution of distributed control systems (DCSs) presents us a lot of data, but also another trouble in nonlinear soft sensing: excessive input variables. If the NN was trained with excessive input variables, the amount of calculations will increase and more computing power is required. In the meantime, the prediction accuracy of the NN is worsened due to extraneous variables creating additional noise in the dataset. Hence, many researchers have focused on the efficiency of variable selection approaches for soft sensors [21–23]. In recent years, mutual information (MI)-based variable selection approaches have been widely studied due to their efficacy and ease of realization [24]. Hanchuan et al. [25] proposed a minimal redundancy maximal correlation criterion to reduce redundancy and apply primitive methods of relevance and redundancy to select significant input variables. Estevez et al. [26] proposed an enhanced version of MIFS and minimal redundancy maximal correlation that imported the normalized MI as an evaluation of redundancy. The developed NMIFS algorithm showed better performance by compensating for the MI partial to multiple features and limiting its setpoint range [0, 1].

This paper develops a new soft sensor algorithm by combining LSTM with NMIFS, in which the NMIFS is used to compress input variables of LSTM. The primary contributions of the paper are summarized as follows:

1. A novel feature selection approach for LSTM with NMIFS is designed. The developed method can effectively reduce the excessive complexity caused by redundant candidate variables and then improve the modelling performance of LSTM.
2. The developed soft sensor algorithm is implemented in two practical industrial processes.
3. Comparative simulation results demonstrate that the developed soft sensor model has better performance and flexibility in performing feedback control.

This paper is arranged as follows: Section 2 presents background theories of the NMIFS and LSTM, and Section 3 describes the development of the presented approach. Section 4 presents the simulation results and an analysis of the developed soft sensor with datasets of actual processes. Some conclusions are given in Section 5.

2. Theoretical Overview

2.1. Input Variable Selection Techniques. The existence of redundant input variables in the training of the NN often complicates the model, deteriorates the accuracy, and even brings about overfitting. The goal of the input variable feature selection (IVFS) is to exactly select \( n \) variables from the initial candidate variable set \( C \) as the input variable set \( S \) in the modelling, where \( C \) includes multifarious input variables of the algorithm. During variable selection, variables that have less influence on target variable will be deleted from \( S \).

The IVFS algorithm can be applied in a variety of means: (1) sequential forward selection, selecting an input from \( C \) to join \( S \) every time until the prediction accuracy of the model is no longer improved; (2) sequential backward selection, where \( S \) initially includes all input variables and input variables are deleted one at a time until the model performance is no longer improved; or (3) global optimization, which finds the optimal solution among all the variable selection approaches.

It can be demonstrated that if there are \( m \) candidate variables in \( C \), the selection of the \( m \) input variables results in \( 2^m - 1 \) subsets in total. \( S \) is hard to be found with large number of candidate variables. Based on this consideration, a statistical indicator to calculate the extent of dependence between input and output variables is selected, and then the input variable before modelling with NNs is selected. This method of separating variable selection procedures from model calibration procedures can produce a more efficient IVFS algorithm, and the resulting \( S \) has wider applicability to different NN algorithms. It is worth noting that the effectiveness of the IVFS approach is based on the statistical standard applied.

MI is considered as an excellent evaluation standard because it is a random measure and does not make assumptions about the structure of the dependencies between variables. MI is also found to be impervious to data transformations and noise.

2.2. Normalized Mutual Information. Suppose that there are two random variables \( X \) and \( Y \), where \( X \) is the input variable and the output variable \( Y \) depends on \( X \). The definition of MI of \( I(X;Y) \) for a continuous variable can be shown as follows [27]:

\[
I(X;Y) = \int \int p(X,Y) \log \frac{p(X,Y)}{p(X)p(Y)} \, dX dY, \tag{1}
\]

where \( p(X,Y) \) is the joint probability density function (PDF) of two variables and \( p(X) \) and \( p(Y) \) are the marginal PDFs of \( X \) and \( Y \). Figure 1 shows the entropy of \( X \) and \( Y \) and its relationship to their MI, in which \( H(X) \) and \( H(Y) \) are entropies, and \( H(X|Y) \) and \( H(Y|X) \) are conditional entropies, respectively.

Generally speaking, MI has three basic attributes:

1. Symmetry: \( I(X;Y) = I(Y;X) \). The quantity of information abstracted from \( Y \) about \( X \) is equal to that
from $X$ about $Y$. The only difference is the angle of the observer.

(2) Positive: $I(X; Y) \geq 0$. Extracting information about one event from another, the worst case is zero information ($I(X; Y) = 0$). Being aware of one event does not strengthen the uncertainty of another.

(3) Extremum: $I(X; Y) \leq H(X)$, $I(Y; X) \leq H(Y)$. The quantity of information abstracted from one case about another is at maximum same as the entropy of the other case, rather than exceeding the amount of information contained by the other event itself.

The MI $I(X; Y)$ provides dependencies for $X$ and $Y$ measurements and provides reference information for variable selection algorithms, which makes the computation of MI a crucial procedure in MI-based input variable selection approaches [28, 29]. However, the mathematical expression form of the PDF in equation (1) is unconscious in practical problems. A variety of approximate prediction algorithms of MI have been extensively researched to analyze PDFs. For example, kernel density estimation (KDE) is an advanced technique that superposes a basis function on each point of the feature data, usually a Gaussian function. The PDF approximation can then be obtained by adopting an envelope of all the basic functions superimposed on each point. Although these kinds of algorithms bring superior approximation results, the computation load is very high, especially in large-scale problems. Histogram methods provide another competitive method, with admissible precision and significantly more computational performance than KDE methods.

When MI is applied to practical cases, the calculation results fluctuate greatly, and it is difficult to directly compare the similarity between several variables and the target variables used as indicators [30]. This paper introduces a method to normalize MI. There are several methods of doing so. The general idea is to use entropy as the denominator to regulate the value of MI to between 0 and 1. One common implementation is the following formula:

$$N(X, Y) = 2 \frac{I(X, Y)}{H(X) + H(Y)}$$

Then, NMI can be used to evaluate the resemblance between candidate and target input variables.

2.3. LongShort-Term Memory. The LSTM network is applied to predicting target variables with relatively long intervals and postponements in the time series. The structure of neurons in LSTM is shown in Figure 2. It includes a cell state and three gate settings: the cell state is used to record neuron status, the input and output gates are used to receive and output parameters, respectively, and the forget gate is used to dominate the degree of forgetting of the previous unit state [31, 32].

The detailed structure and operation mechanism of LSTM are shown in Figure 3. The forgotten part of the memory unit is decided by the input $x_t$ in the forgetting gate together with the state memory unit $S_{t-1}$ and the intermediate output $h_{t-1}$. The retention vector in the memory unit is determined by the changed $x_t$ in the input gate through the sigmoid and tanh functions. The intermediate output $h_t$ is determined by the updated $S_t$ and output $o_t$. The calculation formula is as follows:

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f),$$
$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i),$$
$$g_t = \varnothing(W_{gx}x_t + W_{gh}h_{t-1} + b_g),$$
$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o),$$
$$S_t = g_t \odot i_t + S_{t-1} \odot f_t,$$
$$h_t = \varnothing(S_t) \odot o_t,$$

where $f_t, i_t, g_t, o_t$, and $S_t$ are the states of the forgetting gate, input gate, input node, output gate, intermediate output, and status unit, respectively; $W_{fx}, W_{fh}, W_{ih}, W_{gx}, W_{gh}, W_{ox}$, and $W_{oh}$ are the matrix weight multiplied by input $x_t$ of the corresponding gate and the intermediate output $h_{t-1}$, respectively; $b_f, b_i, b_g$, and $b_o$ are the biases of the corresponding gates; $\varnothing$ indicates that the elements in the vector are multiplied by bits; and $\sigma$ represents the transformation of the sigmoid and tanh function, respectively.

3. Development of NMIFS-LSTM

The evaluation function plays a pivotal role in the MI feature selection, which directly affects the final performance of the algorithm. The method of selecting the variable with the most MI of output variable $Y$ and input variable $X_i$ is the most direct solution. The evaluation function is shown in

$$R = I(X_i; Y).$$

The MIFS [33] method introduces penalty terms based on the measure of relevance, which incorporates correlation and redundancy between variables. The evaluation function is shown in
where $S$ is the selected feature subset, $X_s$ is the selected feature, and parameter $\beta$ controls the degree of penalty for redundant items.

In order to reduce the dependence on parameter $\beta$, Kwak and Chong-Ho Choi [34] proposed the method of MIFS-U, and the evaluation function expression is exhibited as

$$ R = I(X_i; Y) - \beta \sum_{x_{as}} I(X_i; X_s). \quad (5) $$

Hanchuan et al. [25] enhanced MIFS and developed the minimal redundancy maximal relevance algorithm, which establishes a relationship between sample size and parameter $\beta$. The mean value of MI is used as the redundancy evaluation index to avoid the selection of parameter $\beta$. The evaluation function can be shown as

$$ R = I(X_i; Y) - \frac{1}{|S|} \beta \sum_{x_{as}} I(X_i; X_s). \quad (7) $$

The standardized MI between variables was defined by Estevez et al. [26], and the NMIFS algorithm was proposed. Its evaluation function can be expressed as

$$ R = I(X_i; Y) - \frac{1}{|S|} \beta \sum_{x_{as}} NI(X_i; X_s), \quad (8) $$

where the standardized MI is performed as equation (9). The regularized MI compensates for the bias of MI to

**Figure 2: LSTM neuron structure.**

**Figure 3: LSTM internal unit structure.**
multivalued variables, and the regularized MI value is strictly restricted to the interval of \([0, 1]\).

\[
NI(X_i; X_s) = \frac{I(X_i; X_s)}{\min\{H(X_i), H(X_s)\}}.
\] (9)

In this paper, a novel variable selection method of NMIFS-LSTM is developed. This method combines NMIFS and LSTM, and after that, the root mean square error (RMSE) of the LSTM network is used as the evaluation standard. The proposed algorithm aims to eliminate redundant variables and improve model accuracy. The pseudocode of NMIFS-LSTM can be shown in Algorithm 1.

The operating mechanism of the NMIFS-LSTM algorithm is mainly divided into two parts. In the algorithm, we build a model for prediction by LSTM with NMI for variable selection. The LSTM NN is trained to determine network hyperparameters and structure. Parameter \(F\) is set to "initial set of \(n\) variables" and parameter \(S\) set to "empty set." The calculation method is NMI with the LSTM of RMSE and the first variable is chosen. We continue to choose next variables every step until the model gets worse or meets the stop criterion. Finally, the selected subset is modeled and the predicted value is obtained. The flowchart of the developed NMIFS-LSTM-based soft sensor model is shown in Figure 4.

**Algorithm 1: Pseudocode of NMIFS-LSTM.**

| Input: dataset input MT shadow |
| Output: predicted value |
| Begin algorithm |
| Initialize |
| LSTM is trained to determine network hyperparameters and network structure; |
| Set \(F\) = \(n\); \(S\) = empty set (\(n\) = number of input variables); |
| Computation of NMI with LSTM; |
| For \(i = 1:j\) (\(j\) is frequency of the stop criterion) |
| \(\forall f_i \in F\) compute \(I(L; f_i)\); |
| Find a first variable \(f_i\) that maximizes \(I(L; f_i)\) and obtain RMSE; |
| Set \(F \leftarrow F \setminus \{f_i\}\), set \(i \leftarrow 1\); |
| Choose the next variable \(f_i = \arg\max f_i \in F - S(\text{min } \text{fs} \in S(I(f_i, f_i; L)))\) and obtain new RMSE; |
| set \(F \leftarrow F \setminus \{f_i\}\); set \(S \leftarrow S \cup \{f_i\}\); \(j \leftarrow j + 1\); |
| if new RMSE > RMSE |
| Break |
| Else |
| \(\text{RMSE} = \text{newRMSE}\), return and select the next variable; |
| End if |
| End for |
| Retrain with selected subset |
| Calculate predicted value |
| End algorithm |

4. Simulation Results and Discussion

In this paper, all algorithms use a common dataset with the same variable selection method after several trials in the same simulation environment setting. All established models were simulated in the same experimental environment. The program for algorithm simulation was coded in MATLAB 2019 and run under a Windows 8.1 operating system. The simulation results are recorded with the following standards:

1. **Model size (MS)** means the number of candidate variables selected in the ultimate algorithm.
2. **PMSE** means the mean square error (MSE) is a measure that reflects the difference between the actual and value predicted value and can be calculated as follows:

\[
\text{PMSE} = \frac{1}{n_t} \sum_{i=1}^{n_t} (y_i - \bar{y}_i)^2, \tag{10}
\]

where \(y_i\) and \(\bar{y}_i\) are the actual value and predicted value in the algorithm model of the output variable, respectively, and \(n_t\) represents the number of datasets in the testing samples.

3. **Coefficient of determination \((R^2)\)** denotes the square of sample correlation coefficients between the real value and prediction value.

**4.1. Application to a Debutanizer.** To verify the efficacy of the developed soft sensor model, it was applied to a real debutanizer column. The flow diagram of actual debutanizer column unit is given in Figure 5. In the refining industry, the main function of the process is to separate butane from natural gas. At first, the entering liquid is heated into hot steam and then sent into the main tower (T102). The hot vapour condenses into liquid and is separated into a set of fractions with different boiling points. Butane and propane

\[
\text{Input: dataset input MT shadow} \\
\text{Output: predicted value} \\
\text{Begin algorithm} \\
| Initialize |
| LSTM is trained to determine network hyperparameters and network structure; |
| Set \(F\) = \(n\); \(S\) = empty set (\(n\) = number of input variables); |
| Computation of NMI with LSTM; |
| For \(i = 1:j\) (\(j\) is frequency of the stop criterion) |
| \(\forall f_i \in F\) compute \(I(L; f_i)\); |
| Find a first variable \(f_i\) that maximizes \(I(L; f_i)\) and obtain RMSE; |
| Set \(F \leftarrow F \setminus \{f_i\}\), set \(i \leftarrow 1\); |
| Choose the next variable \(f_i = \arg\max f_i \in F - S(\text{min } \text{fs} \in S(I(f_i, f_i; L)))\) and obtain new RMSE; |
| set \(F \leftarrow F \setminus \{f_i\}\); set \(S \leftarrow S \cup \{f_i\}\); \(j \leftarrow j + 1\); |
| if new RMSE > RMSE |
| Break |
| Else |
| \(\text{RMSE} = \text{newRMSE}\), return and select the next variable; |
| End if |
| End for |
| Retrain with selected subset |
| Calculate predicted value |
| End algorithm |
Industrial process

Training time sequences
\((x(t), y(t)), t = 1, 2, \ldots, n\)

Data preprocessing

Training LSTM

Testing

Determine network hyperparameters

Determine network structure

NMIFS-LSTM variable selection

Initialization: Set \(S\) \(\rightarrow\) empty set
Set \(F\) \(\rightarrow\) initial set of \(n\) variables

Computation of the MI with the output class

Choice of the first variable with eq (4), set \(i = 1\)

Run LSTM and get RMSE

Selection of the next variable with eq (8) \(i = i + 1\)

\(\text{RMSE} = \text{newRMSE}\)

Run LSTM again and get newRMSE

\(\text{newRMSE} > \text{RMSE}\) or

stopping criterion: \(i < j\)

Retrain with select subset

Predicted quality value

Figure 4: Flowchart of the proposed NMIFS-LSTM-based soft sensor model.

Figure 5: Brief diagram of debutanizer column.
are detached in the column after the treatment under normal circumstances, which makes the natural gas almost pure methane.

In this case, the content of butane is very important to ensure the product quality during the process. However, this variable is very hard to measure in real time. Hence, a compatible online soft sensing model was proposed to forecast the content. Seven practical sensors were installed in the process, marked as yellow circles in the brief diagram [35], as displayed in Figure 5. All of these candidate variables are listed in Table 1.

2394 data samples were presented at intervals of 15 minutes. The dataset was separated into two parts: the dataset of the first 80% applied for training and the others for testing. On the basis of plant experts guidance [22], the time delay of the process was probably 20–60 minutes. Based on this advice, we extended the input variables to \( x = \{ x_1(t), \ldots, x_p(t), x_1(t-1), \ldots, x_p(t-1), \ldots, x_1(t-4), \ldots, x_p(t-4) \} \), in which \( x_i(t-j) \) means the value of \( x_i \) at time \( t-j \). In addition, we added a guided value of \( (y-t) \) in each group to enhance the accuracy of modelling. The number of candidate variables was raised from 7 to 40, which resulted in additional complexity of the process.

Table 2 presents the experimental results with these four algorithms. The table shows that the NMI-LSTM algorithm has obvious advantage over others in model

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>TI-040</td>
<td>Top temperature</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>PRC-011</td>
<td>Top pressure</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>FRC-015</td>
<td>Reflux flow</td>
</tr>
<tr>
<td>( x_4 )</td>
<td>FRC-018</td>
<td>Flow to next system</td>
</tr>
<tr>
<td>( x_5 )</td>
<td>TRC-004</td>
<td>6th tray temperature</td>
</tr>
<tr>
<td>( x_6 )</td>
<td>TI-036</td>
<td>Bottom temperature 1</td>
</tr>
<tr>
<td>( x_7 )</td>
<td>TI-037</td>
<td>Bottom temperature 2</td>
</tr>
</tbody>
</table>

**Table 2: Statistical results for butane content prediction.**

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>NMI-MLP</th>
<th>LSTM</th>
<th>NMI-LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS</td>
<td>40</td>
<td>12</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>PMSE</td>
<td>0.0720</td>
<td>0.0673</td>
<td>0.0683</td>
<td>0.0421</td>
</tr>
<tr>
<td>( R^2 )</td>
<td>0.9379</td>
<td>0.9524</td>
<td>0.9502</td>
<td>0.9801</td>
</tr>
</tbody>
</table>

**Figure 6: Measured and predicted content of butane.**

The dataset of the first 80% applied for training and the others for testing. On the basis of plant experts guidance [22], the time delay of the process was probably 20–60 minutes. Based on this advice, we extended the input variables to \( x = \{ x_1(t), \ldots, x_p(t), x_1(t-1), \ldots, x_p(t-1), \ldots, x_1(t-4), \ldots, x_p(t-4) \} \), in which \( x_i(t-j) \) means the value of \( x_i \) at time \( t-j \). In addition, we added a guided value of \( (y-t) \) in each group to enhance the accuracy of modelling. The number of candidate variables was raised from 7 to 40, which resulted in additional complexity of the process.

Table 2 presents the experimental results with these four algorithms. The table shows that the NMI-LSTM algorithm has obvious advantage over others in model
accuracy. The simulation performance presents that NMI-LSTM shows a tighter and higher accurate model than other methods.

Figure 6 shows the real values and prediction values of target variable by applying the NMIFS-LSTM model. The fitted graph clearly illustrates that our approach can follow
the variations of the butane content successfully, which further verifies its efficacy.

4.2. Application to Power Plant Desulfurization Technology. The flue gas desulfurization system and industrial process parameters are basically collected from unit 9 of a thermal power plant, which achieves limestone-gypsum wet flue gas desulfurization technology with twin towers. The system’s SO₂ is absorbed by lime or limestone with chemical reaction. Compared to the single tower, these twin towers can carry out secondary reaction of transmitted flue gas and eliminate SO₂ in the flue gas more successfully. The flue gas desulfurization process includes SO₂ absorption system, flue gas system, mist eliminator system, absorption tower overflow device, slurry mixing system of absorption tower, oxidizing blower, etc. These twin towers have an absorption area of 12 meters in diameter and a height of 32.6 meters. The flue gas containing SO₂ moves from bottom to top where the bottom of the primary absorption tower (PAT) and encounters a liquid suspension from the spray layer. SO₂ chemically reacts with the alkaline suspension through the gas film and the liquid film in a molecular diffusion manner. The PAT includes four spray layers that are dominated by circulating pumps, shown in Figure 7.

This paper collects the data sample of desulfurization index parameters of unit 9 of a thermal power plant as the research object. The dataset includes 30 input variables and a target output variable flue gas SO₂ concentration. All candidate variables are given in Table 3. The time span is from July 1, 2019, to July 7, 2019, with a time interval of 1 min and a total of 10000 samples. The first 8000 samples are used as the training data and the others are used as the testing data. In the practical simulation experiment, redundant variables in the pool of candidate variables can lead to unsuitable modelling. Consequently, IVFS technique is very important for building a suitable and stable soft measurement model.

Table 4 presents the statistics of data-driven models with different algorithms. Experimental results present that NMI-LSTM has better performance with fewer input variables than other approaches. $R^2$ of NMI-LSTM is higher than 90%, representing that the proposed soft sensor can precisely forecast the actual values.

Figure 8 shows the prediction curve of SO₂ concentration by NMI-LSTM algorithm. Obviously, NMI-LSTM can

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>NMI-MLP</th>
<th>LSTM</th>
<th>NMI-LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS</td>
<td>30</td>
<td>11</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td>PMSE</td>
<td>5.9569</td>
<td>3.3295</td>
<td>3.2419</td>
<td>2.1233</td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.3392</td>
<td>0.7765</td>
<td>0.7524</td>
<td>0.9023</td>
</tr>
</tbody>
</table>

**Figure 8:** Prediction curve of flue gas SO₂ concentration by NMI-LSTM algorithm.
track the dynamic change of target variable effectively, which shows that our algorithm is very effective.

5. Conclusion

In this paper, a novel soft sensor was designed to model complex and dynamic industrial processes with time series characteristics. The LSTM network is trained by datasets taken from actual processes, and NMI is applied to select the variables related to the target variable. The proposed algorithm deletes one irrelevant variable at every step until all the variables are removed. After that, the path of variable selection appears and the algorithm takes the segment with the lowest prediction error. The proposed soft sensor was applied to two practical industrial processes. The simulation and comparison with other algorithms demonstrate the effectiveness and excellence of our approach. The developed soft sensor provides an additional and reliable monitoring tool for pivotal variables and can be further applied to the design of model predictive control systems.

The proposed soft sensor algorithm is easy to implement, and the related program can be preserved as a subroutine in the industrial computer of the DCS. By calling the subroutine, the soft sensor could be periodically retrained and updated with the new production data. The disadvantage of model degradation can be completely eliminated with this technique.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors’ Contributions

Dongfeng Li and Zhirui Li contributed equally to this article.

Acknowledgments

The authors acknowledge the Yau Mathematical Sciences Center and Huangtai Power Plant for data gathering and experiment. This work was partially supported by the Major Science and Technology Innovation Projects of Shandong Province (Grant no. 2019ZYY010731), the Key Research and Development Program of Shandong Province (Grant no. 2019GGX104037), the S.-T. Yau High School Science and Development Program of Shandong Province (Grant no. 2019JZZY010731), the Key Research and Technology Innovation Projects of Shandong Province (Grant no. 2019GGX104037), the S.-T. Yau High School Science Award (Computer), and the National Natural Science Foundation of China (Grant no. 51874300).

References


