2021 IEEE 10th Data Driven Control and Learning Systems Conference
May 14-16, 2021, Suzhou, China

A new variable selection algorithm for LSTM neural network

Lin Sui[1], Bosheng Du[1], Mengyan Zhang[1], Kai Sun[1]

1. School of Electrical Engineering and Automation, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
E-mail: sunkai79@qlu.edu.cn

Abstract: This paper proposes an accurate and reliable input variable selection algorithm by embedding a nonnegative garrote (NNG) algorithm into long short term memory (LSTM) neural network to perform data-driven modeling on a highly nonlinear and dynamic time-delay dataset. Firstly, an LSTM deep neural network is trained, and a well-trained LSTM network is obtained by optimizing the parameters of LSTM through a grid search algorithm. Secondly, the initial input weights of LSTM are compressed accurately by the NNG algorithm, and block cross-validation is applied to the optimization calculation process to achieve input variable selection. Finally, the performance of the algorithm is verified by the improved Friedman time-delay artificial datasets. Simulation results show that the algorithm could construct a more simplified and better predictive model than other traditional algorithms.

Key Words: LSTM, Variable selection, Nonnegative garrote, Time-series, Dynamic

2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS) | 978-1-6654-2423-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/DDCLS52934.2021.9455564

1 Introduction

With the development of modern industrial intelligence and the popularity of distributed control systems, many industrial process data are collected and stored, which provides the necessary conditions for the development of data-driven modeling technology [1]. Typical data-driven modeling methods include principal component analysis (PCA) [2], partial least squares (PLS) [3], support vector machine (SVM) [4] and artificial neural network (ANN), etc. Among them, the ANN has been widely used in data-driven modeling on complex processes, owing to its very powerful capability of nonlinear mapping and self-learning [5-7].

However, the actual industrial data is generally highly nonlinear and dynamic, with strong temporal correlations between data samples. Ordinary static ANN is difficult to capture the complex dynamic correspondence between time series data. Hence, researchers focused on studying recurrent neural networks (RNN) that are robust for nonlinear dynamic statistical data modeling. The RNN can effectively describe temporal dynamic behavior for time sequences by introducing time series feedback mechanism, considering the current process state and relevant historical information [8, 9]. However, the standard RNN suffers from gradient vanishing and explosion problems when models time sequences with long time intervals and delay. Therefore, Hochreiter et al. [10] developed a long short-term memory (LSTM) neural network as an improved version of RNN to deal with this problem. Compared with standard RNN, LSTM has a more complex hidden layer structure, and the advanced convergence method thus has a more vital historical information processing ability. In recent years, LSTM is becoming increasingly popular in many fields, such as image classification [11], natural language processing [12], and complex system modeling [13].

Although LSTM is excellent in predicting time sequences with long time intervals and delay, the available data sets of practical problems usually have a large number of candidate explanatory variables with high cross-correlation. Redundant input variables of the process increase the computational complexity and worsen the accuracy of the model. Variable selection techniques are useful means to decrease the complexity of the models and enhance their performance. Many researchers are focusing on the development of efficacious variable selection approaches for LSTM neural network-based inferential models in recent years. Yang et al. [14] proposed a predictive model based on mutual information variable selection and LSTM neural network to achieve dynamic forecasting for NOx emission. The experimental results showed that the proposed method decreased the number of input variables and improved the accuracy and robustness of the model. Sun et al. [15] effectively selected feature variables according to the criterion of maximizing the relevancy and minimizing the redundancy, and then a deep learning LSTM neural network is proposed to predict the load consumption. This novel strategy captures distinct load characteristics, chooses accurate input variables, and presented excellent forecasting. Yuan et al. [16] proposed an LSTM neural network soft sensing algorithm based on spatiotemporal attention, by applying the attention mechanism to obtain the spatial correlation between the input and the target variables and then selecting the critical input variables related to the quality variables.

Last few years, coefficient shrinkage algorithms with penalty likelihood functions had been widely studied. Sun et al. [17] proposed an innovative variable selection algorithm for soft sensor using multi-layer perceptron (MLP) and nonnegative garrote (NNG), and simulation results demonstrated the efficacy and superiority of the algorithm. However, the model is built based on the steady-state and static assumptions of the process, which only considers the instantaneous corresponding relationship between the target variables, and the input variables and ignores the time-delay characteristics and dynamic nature of the practical process.

*This work is partially supported by the Key Research and Development Program of Shandong Province (Grant No. 2019GGX104037), and the Major Science and Technology Innovation Projects of Shandong Province (Grant No.2019JZZY010731).

978-1-6654-2423-3/21/$31.00 ©2021 IEEE

Therefore, this paper proposes a variable selection algorithm for LSTM neural network by combining the NNG algorithm with LSTM neural network. The proposed algorithm utilizes the dynamic feature extraction capability of the LSTM neural network to exploit the complex time series correspondence between auxiliary variables and target variables, and uses penalty functions to select input variables and improves the generalization performance of the model.

The rest of this article is organized as follows. In Section 2, the model structure of RNN and LSTM is reviewed. The variable selection for the LSTM neural network modeling is designed and analyzed in section 3. In Section 4, simulation results are given, and the algorithm's effectiveness is verified on artificial data set. Some conclusions are made at last.

2 Overview of RNN and LSTM structure

2.1 Recurrent neural network

RNN is an artificial neural network with historical information memory ability, mainly composed of the input layer, hidden layer, and output layer. The key to its memory ability lies in the circular connection between the hidden layers. That is, the hidden layer information is not only passed to the output layer but also used as the input of the hidden layer at the next moment. RNN is a chain network structure with repeated modules, and the chain expansion structure of its adjacent nodes is shown in Fig. 1.

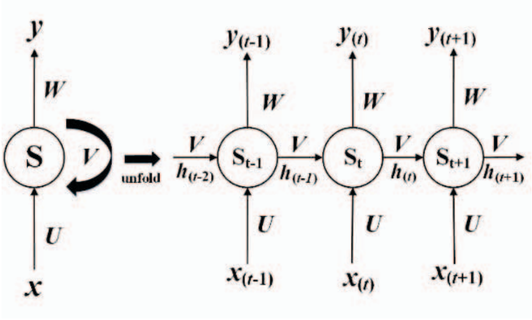

Fig. 1: Chain deployment structure of RNN adjacent nodes

The hidden layer of the traditional RNN only contains a simple activation function (such as $tanh(\cdot)$, etc.), and its basic operating principle can be described as:

$$h_{(t)} = f\left(Ux_{(t)} + Vh_{(t-1)} + b\right) \tag{1}$$

$$y_{(t)} = \mathrm{g}\left(Wh_{(t)} + \mathrm{c}\right) \tag{2}$$

where $x_{(t)}$, $h_{(t)}$ and $y_{(t)}$ represent the input, implied and output vectors of RNN at the current sampling time t, *f(·)* and *g(·)* are the activation functions, *U*, *V* and *W* are the corresponding connection weights, and *b* and *c* are the bias vectors.

Theoretically, the multilayer structure of the RNN hidden layer allows the persistence of information. However, RNN will face the problem of gradient disappearance or explosion during training, and it is difficult to pass the information of earlier time step to the time step in the later stage, and it cannot establish the long time dependency between variables.

2.2 Long short-term memory neural network

LSTM is a unique RNN structure, which is proposed to overcome the problem of the long-term dependence in traditional RNN. LSTM neural network uses information storage units instead of RNN implicit neurons to realize long-term memory of information. The core of the LSTM information storage unit lies in its cell state and unique gating unit structure: the cell state is equivalent to the information transmission path and is responsible for passing relevant information in the sequence processing to the next moment; the gating unit is responsible for regulating the flow of information, make the network selectively forget, save or add certain related information. The detailed structure of the LSTM information storage unit is shown in Fig. 2.

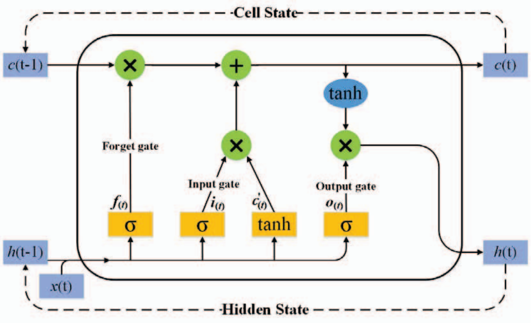

Fig. 2: LSTM information storage unit

For the basic LSTM unit, its external inputs are its previous cell state c_{t-1}, the previous hidden state h_{t-1}, and the current input vector x_t. Its two outputs are the cell state c_t and the hidden state h_t at the current moment, and they are used as the unit input at the next moment to participate in the calculation. The gating mechanism of LSTM is composed of forget gate (f_t), input gate (i_t), output gate (o_t) and temporary storage state (c_t'), which together determine the forgetting, retention or addition of relevant information in the cell state (c_t).

The f_t is used to control discarding or continuing to save the message at the previous moment in the storage state c_t; i_t and c_t jointly determine how much information of the current input x_t is saved to c_t; o_t is used to determine which relevant information c_t needs to output to h_t . The calculation process of the LSTM gating mechanism is as follows:

$$f_t = \sigma\left(W_{xf}x_t + W_{ht}h_{t-1} + b_f\right) \tag{3}$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{4}$$

$$o_t = (W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{5}$$

$$c_t' = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{6}$$

in which, f_t, i_t, o_t and c_t' represent the output of forgetting gate, input gate, output gate and temporary storage state respectively. Following that, the output of cell state(c_t) and hidden layer state(h_t) are updated as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot c_t' \tag{7}$$

$$h_t = o_t \odot tanh\,(c_t) \tag{8}$$

where b is the bias value corresponding to each gate, W_x represents the weight of each gate related to the input information x, W_h means the weight of each gate related to the output h_{t-1} at the previous moment. The $\sigma(\cdot)$ and $tanh(\cdot)$ denote sigmoid nonlinear activation function and hyperbolic tangent activation function respectively, and ⊙ denotes multiply point by point.

Finally, the equation of the output vector at sample time$t$ is as follows:

$$y_t = \mathrm{g}\left(W_y\left(o_t \odot \tanh\left(f_t \odot c_{t-1} + i_t \odot c_t'\right)\right) + b_y\right) \quad (9)$$

3 Proposed methodology

3.1 Nonnegative Garrote

NNG is a variable selection algorithm with penalty constraints, which has excellent coefficient shrinking ability and was first used to solve linear subset regression problem:

$$y = x\beta + \varepsilon \quad (10)$$

where $x = \left[x_1, x_2, \ldots, x_p\right]$ and y represent input and output variables respectively, $\beta = \left[\beta_1, \beta_2, \ldots, \beta_p\right]^T$ is the coefficient matrix, and ε is the random error. Breiman designed the contraction vector $\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2, \ldots, \alpha_p\right]$, and exerted it to the ordinary least squares (OLS) regression expression:

$$\begin{cases} \boldsymbol{\alpha}^*(s) = \underset{\forall(x,y)\in\{X,Y\}}{\operatorname{argmin}} \left\{\sum_{k=1}^{n}\left(y_k - \sum_{i=1}^{p} \alpha_i x_{ik} \hat{\beta}_i\right)^2\right\} \\ \quad s.t.\ \ \alpha_i \geq 0, \sum_{i=1}^{p} \alpha_i \leq s \end{cases} \quad (11)$$

where $\hat{\beta}_i$ denotes coefficient vector of OLS estimate and s is the garrote parameter. X∈ $\mathbb{R}^{n \times p}$ is the input data matrix, in which each column signifies a candidate input variable and n represents the number of samples in the data matrix. Y∈ $\mathbb{R}^n$ is the data matrix of output variable.

For a given value of s, the optimized contraction vector $\boldsymbol{\alpha}^* = \left[\alpha_1^*, \alpha_2^*, \ldots, \alpha_p^*\right]$ is obtained by solving Eq. (11). After that, $\boldsymbol{\alpha}^*$ is substituted into Eq. (10) to present an optimized coefficient estimate:

$$\tilde{\beta}_i = \boldsymbol{\alpha}^* \hat{\beta}_i, i = 1, 2, \cdots, p \quad (12)$$

for any variable, the corresponding variable x_i is deleted from the set of input variables if α_i^*=0. The predictive model of y can be presented as:

$$\tilde{y} = \sum_{i=1}^{p} \tilde{\beta}_i x_i \quad (13)$$

In the NNG algorithm, the value of s directly determines the strangulation strength of the algorithm. When the garrote parameter s ≥p, the constraint $\sum_{i=1}^{p} \alpha_i \leq s$ of Eq. (11) is inactive. In this situation, $\tilde{\beta}_i = \hat{\beta}_i$, and all input variables will be preserved. With the decrease of s, the strangulation intensity of the algorithm is enhanced. More α_i^* tends to zero under this condition, meaning that more variables will be eliminated. When s decreases to zero, all input variables are eliminated and a null model is presented. The algorithm regulates the strangulation intensity by adjusting the value of s, and selects the best strangulation parameters and the corresponding model according to the model selection criteria.

3.2 Integrate the NNG algorithm into LSTM

In this paper, the algorithm of NNG is introduced as a penalty function into the LSTM network in order to obtain a more simplified model with input variable selection. The physical architecture of LSTM is demonstrated in Fig. 3.

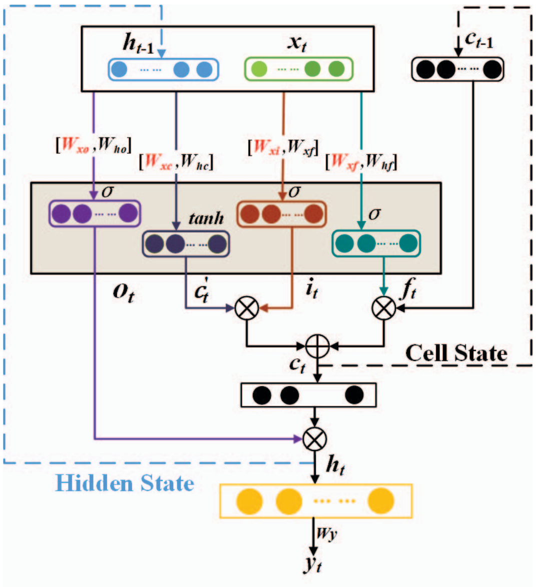

Fig. 3: Physical architecture of LSTM

The algorithm of NNG-LSTM includes two stages: in the first stage, the grid search method is used to optimize the hyper-parameters of the LSTM neural network. A well-trained LSTM neural network is obtained as the initial model, as shown in Eq. (9). In the second stage, the contraction coefficient $\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2, \ldots, \alpha_p\right]$ of the NNG algorithm is added to the LSTM network input weight, and a new LSTM neural network expression is established:

$$\hat{y}_t = \mathrm{g}\left(W_y\left(\hat{o}_t \odot \tanh\left(\hat{f}_t \odot c_{t-1} + \hat{\imath}_t \odot \hat{c}_t'\right)\right) + b_y\right) \quad (14)$$

where the output of each gate control unit is updated as follows:

$$\hat{f}_t = \sigma\left(W_{xf}(\alpha \cdot x_t) + W_{ht} h_{t-1} + b_f\right) \quad (15)$$

$$\hat{\imath}_t = \sigma\left(W_{xi}(\alpha \cdot x_t) + W_{hi} h_{t-1} + b_i\right) \quad (16)$$

$$\hat{o}_t = \sigma\left(W_{xo}(\alpha \cdot x_t) + W_{ho} h_{t-1} + b_o\right) \quad (17)$$

$$\hat{c}_t' = tanh\left(W_{xc}(\alpha \cdot x_t) + W_{hc} h_{t-1} + b_c\right) \quad (18)$$

then, the NNG-LSTM algorithm can be expressed as:

$$\begin{cases} \boldsymbol{\alpha}^*(s) = argmin\left\{\sum_{k=1}^{n}(y_k - \hat{y}_t)^2\right\} \\ \quad s.t.\ \ \alpha_i \geq 0, \sum_{i=1}^{p} \alpha_i \leq s \end{cases} \quad (19)$$

It is very clear that the equation (19) is a quadratic minimization problem with nonlinear constraints. For a given strangulation parameter s, the trust region algorithm [18, 19] can be used to solve the problem to get the optimal contraction vector $\boldsymbol{\alpha}^*$. This approach generates strictly feasible iterates by using a new affine scaling transformation and following piecewise linear paths (reflection paths). There is no stability problem, and computation time increases only moderately as the number of input variables increases.

By substituting $\boldsymbol{\alpha}^*$ into Eq. (19), a new set of input weights $\widetilde{W}_{ik}$ can be obtained by:

$$\widetilde{W}_{ik} = \boldsymbol{\alpha}^* \odot \widehat{W}_{ik},\ i = 1, 2, \ldots, p;\ \ k = 1, 2, \ldots, q \quad (20)$$

where $\boldsymbol{\alpha}^* = \left[\alpha_1^*, \alpha_2^*, \ldots, \alpha_p^*\right]$, for any variable. If α_i^*=0, the input variable x_i^t will be deleted, and the output of each gating unit is updated as:

$$\tilde{f}_t = \sigma\left(\widehat{W}_{xf} x_t + W_{ht} h_{t-1} + b_f\right) \quad (21)$$

$$\tilde{\imath}_t = \sigma\left(\widehat{W}_{xi} x_t + W_{hi} h_{t-1} + b_i\right) \quad (22)$$

$$\tilde{o}_t = \sigma\left(\widehat{W}_{xo} x_t + W_{ho} h_{t-1} + b_o\right) \quad (23)$$

$$\tilde{c}_t' = tanh\left(\widehat{W}_{xc} x_t + W_{hc} h_{t-1} + b_c\right) \quad (24)$$

Then, the output variable $\hat{y}_t$ of the optimized LSTM model is presented as:

$$\hat{y}_t = \mathrm{g}\left(W_y\left(\tilde{o}_t \odot \tanh\left(\tilde{f}_t \odot c_{t-1} + \tilde{\imath}_t \odot \tilde{c}_t'\right)\right) + b_y\right) \quad (25)$$

3.3 Determination of parameters

Grid search (GS) is an exhaustive search strategy with specified parameter values, which has the advantages of simple and practical, parallel computing, and controllable time-consuming [20]. It is an effective method to solve the problem of model hyperparameters optimization.

In this paper, the GS strategy is applied to optimize the hyperparameters of the LSTM network. Firstly, the candidate confidence values of different hyperparameters are determined according to prior knowledge, and the candidate hyperparameters grid is generated. Secondly, the GS method is used to optimize the combination of different super parameters, and the optimal hyperparameters combination is chosen. Finally, the guided initial LSTM network is obtained according to the optimized hyperparameters combination.

In this algorithm, the choice of garrote parameter is essential to the performance due to its direct influence on the complexity and prediction accuracy of the final model. In order to select the optimal parameter, the value of s is determined by enumeration in the linear bisection vector S_l=[s_1, s_2,…,s_u], where the $s_1 = 0$ is the lower bound, $s_u = p$ is the upper bound, u is the size of the vector, and Δs is the difference between adjacent elements. The optimal value of s was obtained by cross-validation (CV) evaluation.

The CV is a standard model evaluation method to adjust hyperparameters of the model, which can evaluate the generalization ability of model by predicting results on other independent data sets. However, the ordinary CV method divide the data set in a random way, which will undermine the time dependence of the data and produce a biased estimation[21,22]. Therefore, this paper uses blocked cross-validation (BCV) [23] to evaluate model performance and determine the optimal value of the strangulation hyper parameter s for time series data, and the schematic diagram of the approach is shown in Fig. 4.

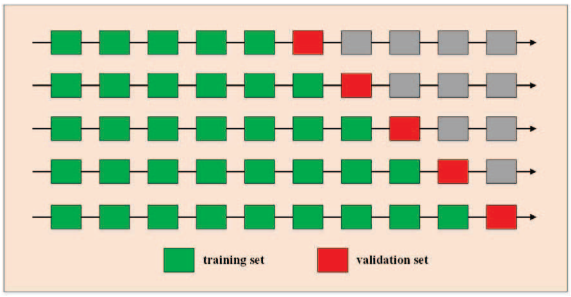

Fig. 4: The diagram of blocked cross-validation

The key to BCV is to ensure the time dependence between the subset blocks when dividing the time series data set. The forward link strategy is used to divide the data set into V block subsets, in other words, divide each block subset in chronological order and then gradually select new ones.

3.4 The execution process of BCV

The execution process of BCV is as follows:

The dataset $\delta = \{X, Y\}$ is divided into K subsets: $\delta = \{\delta_1, \delta_2, \dots, \delta_J, \dots, \delta_K\}$, the training set is $\overline{\delta_J} = \{\delta_1, \delta_2, \dots, \delta_J\}$ and the validation set is $\widetilde{\delta_J} = \{\delta_{J+1}\}$. Then gradually scroll forward and select the new validation set $\overline{\delta_J} = \{\delta_1, \delta_2, \dots, \delta_J, \delta_{J+1}\}$ and the validation set $\widetilde{\delta_J} = \{\delta_{J+2}\}$, until the end of K-J fold cross-validation. The NNG-LSTM uses the dataset $\overline{\delta_J}$ establish an appropriate model and the validation dataset $\widetilde{\delta_J}$ calculate the validation value $\tilde{y}^{(J)}$, and mean square error between $\tilde{y}^{(J)}$ and $y^{(J)}$ can be calculated :

$$MSE^J = \frac{1}{n_i}\sum_{i=1}^{n_i}\left(\tilde{y}_i^{(J)} - \tilde{y}_i^{(J)}\right)^2 \quad (26)$$

Perform K-J fold cross verification under the current strangulation parameter s, and calculate the average value. The calculation method is as follows:

$$\overline{BCV}_{(S)}^{K-J} = \frac{1}{K-J}\sum_J^K MSE^J \quad (27)$$

The optimal value s^* of the strangulation parameter can be obtained by minimizing $\overline{BCV}_{(S)}^{K-J}$, and its calculation method is as follows:

$$s^* = argmin_{(s_{lb}<s<s_{ub})}\left(\frac{1}{K-J}\sum_J^K MSE^J\right) \quad (28)$$

4 Simulation result

In this section, the NNG-LSTM algorithm proposed in this paper is evaluated using numerical examples and compares its performance with MLP, NNG-MLP, and LSTM algorithms. The experiment study demonstrates the effectiveness of the proposed method.

4.1 Evaluation criterion

In the paper, the algorithm performance is assessed with the following indicators:

(1) Model size (M.S): the number of input variables remained in the ultimate model.

$$\mathrm{M.S.} = \sum_{i=1}^{n_i} S_i, S_i = \begin{cases} 1, \alpha_i \geq 10^{-5} \\ 0, \alpha_i \leq 10^{-5} \end{cases} \quad (29)$$

where S_i represents whether each input variable is the final member of the model, and α_i is the shrinkage coefficient of each variable.

(2) Correct ratio (C.R): the ratio of correct variables selected.

$$\mathrm{C.R} = \frac{n_c}{\mathrm{M.S}} \times 100\% \quad (30)$$

where n_c is the number of correct variables in M.S.

(3) Coefficient of determination (R^2): The trend of the two sets of data changing and moving together is described, which measures how well the predicted value fits the true value.

$$R^2 = 1 - \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n (y_i - \bar{y}_i)^2} \quad (31)$$

(4) Root mean square error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (32)$$

where y_i and $\hat{y}_i$ are the actual and predicted value of the output variable respectively, $\bar{y}_i$ is the average value of y_i, and n is the number of data samples in the testing dataset.

4.2 Simulation example of artificial dataset

To investigate the performance of the developed NNG-LSTM algorithm, a dataset with time-delay dynamic characteristics is generated based on the Friedman dataset[24]. In this example, the input dataset $X_1 \in \mathbb{R}^{n\times 5}$ and output dataset $Y \in \mathbb{R}^{n\times 1}$ are generated, in which the independent variable is randomly generated, and the corresponding response is assigned by as follows:

$$\begin{cases} Y^t = 10\sin(\pi x_1^t) + 15\sqrt{x_2^t} + 20(x_3^t + 0.1)^2 + 10x_4^T \\ \qquad\qquad +5x_5^T + \varepsilon \\ s.t.\ x_4^T = x_4^{t-1} + x_4^{t-3} + x_4^{t-5},\ \ x_5^T = x_4^{t-2} + x_4^{t-4} \end{cases} \tag{33}$$

where t represents the time relationship between the input variables, x_4^T and x_4^T are input variable with time-delay characteristics, and ε is Gaussian noise. At the same time, to further prove the effectiveness of the algorithm, 25 redundant inputs variable $X_2 \in \mathbb{R}^{n\times 25}$ is added to verify the validity of the variable selection of the algorithm. Therefore, the input variables are $X^T = \{X_1, X_2\}$.

4.3 Experimental setting

In this example, a total of 2000 data are generated, 80% of which is used for training, and the rest is used for testing. All algorithms are executed in the same simulation environment, and the parameter settings of the network are consistent. The numerical results of the simulation with different algorithms are provided in Table 1.

Table 1: Numerical results with different algorithms for data sets of dynamic time series

	MLP	LSTM	NNG-MLP	NNG-LSTM
M.S	30	30	16	12
C.R	16.67%	16.67%	18.75%	41.67%
R^2	0.5036	0.7663	0.6801	0.8714
MSE	50.6820	23.9559	31.7788	14.3438

Table 1 shows that the MSE and M.S of the NNG-LSTM algorithm are significantly lower than other algorithms, while C.R and R^2 are significantly improved. The result means that the proposed algorithm can effectively shrink input variables and obtain more accurate predictive models.

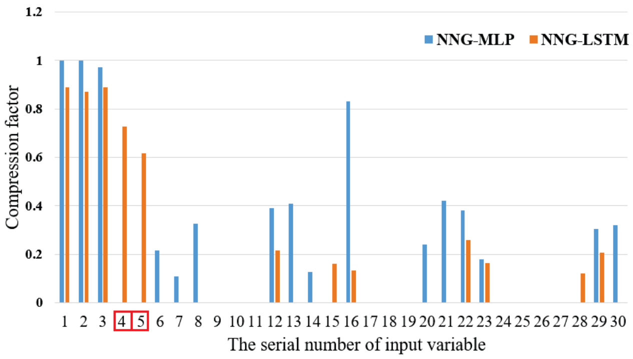

Fig. 5: Compression factor of input variables with different algorithms

Fig. 5 presents the compression factor of input variables with two different algorithms. It can be seen that the coefficients of most redundant variables are effectively compressed, and the first five valid variables are successfully saved with NNG-LSTM algorithm. In contrast, the input variable x_4^T and x_5^T with time-delay characteristics are not recognized in the algorithm of NNG-MLP, which caused them to be deleted and affects the modeling effect. Compared with the MLP that is a typical static neural network, the LSTM neural network can better capture the delay and dynamic characteristics of the process.

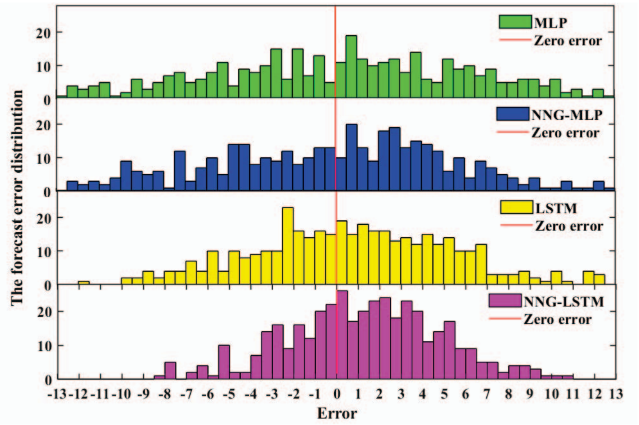

Fig. 6: Histogram of prediction error distribution with different algorithms

Fig. 6 presents the distribution of prediction errors with different algorithms. The comparative results show that the measurement error of the NNG-LSTM algorithm is closer to the normal distribution, and the error fluctuation is the smallest. Our approach has higher prediction accuracy and generalization performance.

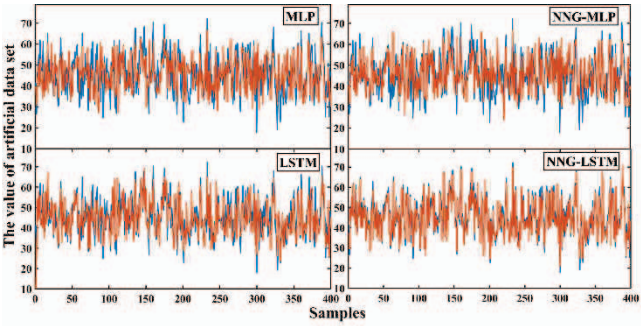

Fig. 7: Comparison of prediction results with different algorithms

Fig. 7 shows the model fitting results on 400 test samples with different algorithms, in which the blue line is the real value and the red line is the prediction value. Obviously, the NNG-LSTM algorithm has the best fitting effect and can successfully follow the dynamic changes of the sample data, which further demonstrates the superiority of the algorithm.

5 Conclusions

In the paper, the NNG is combined into LSTM to develop a novel variable selection algorithm for nonlinear model on high dynamic process. The proposed algorithm makes full use of the long-term historical information memory ability of the LSTM neural network to exploit the complex time series correspondence between auxiliary variables and dominant variables and improves the dynamic information processing performance. On the other hand, the NNG algorithm is used to optimize the LSTM input weight matrix and reduce redundant information among feature variables to improve the generalization performance of the model. Finally, the approach is applied to an artificial dataset with time-delay characteristics. The developed NNG-LSTM is compared to other classical algorithms, and the results show that the proposed algorithm can obtain better prediction models with fewer input variables.

References

[1] D. Voukantsis, H. Niska, K. Karatzas, M. Riga, A. Damialis, and D. Vokou, "Forecasting daily pollen concentrations using data-driven modeling methods in Thessaloniki, Greece," *Atmospheric Environment,* vol. 44, no. 39, pp. 5101-5111, 2010.

[2] T. Loutas, N. Eleftheroglou, G. Georgoulas, P. Loukopoulos, D. Mba, and I. Bennett, "Valve Failure Prognostics in Reciprocating Compressors Utilizing Temperature Measurements, PCA-Based Data Fusion, and Probabilistic Algorithms," *IEEE Transactions on Industrial Electronics,* vol. PP, no. 99, pp. 1-1, 2019.

[3] Q. Jiang, X. Yan, H. Yi, and F. Gao, "Data-Driven Batch-End Quality Modeling and Monitoring Based on Optimized Sparse Partial Least Squares," *IEEE Transactions on Industrial Electronics,* vol. PP, no. 99, pp. 1-1, 2019.

[4] H. Kaneko and K. Funatsu, "Application of online support vector regression for soft sensors," *Aiche Journal,* vol. 60, no. 2, 2014.

[5] K. Sun, S. H. Huang, S. H. Wong, and S. S. Jang, "Design and Application of a Variable Selection Method for Multilayer Perceptron Neural Network With LASSO," *IEEE Transactions on Neural Networks & Learning Systems,* pp. 1-11, 2016.

[6] J. C. B. Gonzaga, L. A. C. Meleiro, C. Kiang, and R. M. Filho, "ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process," *Computers & Chemical Engineering,* vol. 33, no. 1, pp. 43-49, 2009.

[7] M. Karthikeyan, K. Sharmilee, P. M. Balasubramaniam, N. B. Prakash, and S. Sudhakar, "Design and Implementation of ANN-based SAPF Approach for Current Harmonics Mitigation in Industrial Power Systems," *Microprocessors and Microsystems,* p. 103194, 2020.

[8] Z. Zhang and Z. Yan, "An Adaptive Fuzzy Recurrent Neural Network for Solving the Nonrepetitive Motion Problem of Redundant Robot Manipulators," *IEEE Transactions on Fuzzy Systems,* vol. 28, no. 4, pp. 684-691, 2020.

[9] P. Coulibaly, F. Anctil, P. Rasmussen, and B. Bobée, "A recurrent neural networks approach using indices of low - frequency climatic variability to forecast regional annual runoff," *Hydrological Processes,* vol. 14, no. 15, 2000.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[11] A. A. Nahid, M. A. Mehrabi, and Y. Kong, "Frequency-domain information along with LSTM and GRU methods for histopathological breast-image classification," in *IEEE International Symposium on Signal Processing & Information Technology,* 2017.

[12] T. Tanaka, T. Moriya, T. Shinozaki, S. Watanabe, and K. Duh, "Evolutionary optimization of long short-term memory neural network language model," *The Journal of the Acoustical Society of America,* vol. 140, no. 4, pp. 3062-3062, 2016.

[13] X. Yuan, L. Li, and Y. Wang, "Nonlinear Dynamic Soft Sensor Modeling With Supervised Long Short-Term Memory Network," *IEEE Transactions on Industrial Informatics,* pp. 3168-3176, 2020.

[14] G. Yang, Y. Wang, X. Li, and K. Liu, " Dynamic Prediction of Boiler NO_x Emission Based on Mutual Information Variable Selection and LSTM," *Journal of North China Electric Power University(Natural Science Edition)*, vol. v.47;No.205, no. 03, pp. 70-78, 2020.

[15] G. Sun, C. Jiang, X. Wang, and X. Yang, "Short - term building load forecast based on a data_._ining feature selection and LSTM ln NN method," *IEEJ Transactions on Electrical and Electronic Engineering,* vol. 15, no. 7, 2020.

[16] X. Yuan, L. Li, Y. Shardt, Y. Wang, and C. Yang, "Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development," *IEEE Transactions on Industrial Electronics,* 2020.

[17] K. Sun, J. Liu, J. L. Kang, S. S. Jang, S. H. Wong, and D. S. Chen, "Development of a variable selection method for soft sensor using artificial neural network and nonnegative garrote," *Journal of Process Control,* vol. 24, no. 7, pp. 1068-1075, 2014.

[18] T. F. Coleman and Y. Li, "On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds," *Mathematical Programming,* vol. 67, no. 1-3, pp. 189-224, 1994.

[19] T. F. Coleman and Y. Li, "An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds," *Siam Journal on Optimization,* vol. 6, no. 2, pp. 418-445, 1993.

[20] S. M. Lavalle, M. S. Branicky, and S. R. Lindemann, "On the Relationship between Classical Grid Search and Probabilistic Roadmaps," *The International Journal of Robotics Research,* vol. 23, no. 7-8, pp. 673-692, 2004.

[21] G. Jiang and W. Wang, "Markov cross-validation for time series model evaluations," *Information Sciences,* vol. 375, pp. 219-233, 2017.

[22] Jeff and Racine, "Consistent cross-validatory model-selection for dependent data: hv-block cross-validation," *Journal of Econometrics,* 2000.

[23] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences,* vol. 191, no. none, pp. 192-213, 2012.

[24] J. H. Friedman, "Multivariate Adaptive Regression Splines," *Annals of Statistics,* vol. 19, no. 1, pp. 1-67, 1991.