Short-term traffic flow prediction based on whale optimization algorithm optimized BiLSTM_Attention

Author(s): Xu, Xing; Liu, Chengxing; Zhao, Yun; Lv, Xiaoshu

Title: Short-term traffic flow prediction based on whale optimization algorithm optimized BiLSTM_Attention

Year: 2022

Version: Accepted manuscript

Please cite the original version:

Short-term traffic flow prediction based on whale optimization algorithm optimized BiLSTM_Attention

Xing Xu[1] Chengxing Liu[1] Yun Zhao[2] Xiaoshu Lv[3,4]

[1] School of Mechanical and Energy Engineering, Zhejiang University of Science and Technology, Hangzhou, China

[2] School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou, China

[3] Department of Electrical Engineering and Energy Technology, University of Vaasa, Vaasa, Finland

[4] Department of Civil Engineering, Aalto University, Espoo, Finland

Abstract

With the growths in population and vehicles, traffic flow becomes more complex and uncertain disruptions occur more often. Accurate prediction of urban traffic flow is important for intelligent decision-making and warning, however, remains a challenge. Many researchers have applied neural network methods, such as convolutional neural networks and recurrent neural networks, for traffic flow prediction modeling, but training the conventional network that can obtain the best network parameters and structure is difficult, different hyperparameters lead to different network structures. Therefore, this article proposes a traffic flow prediction model based on the whale optimization algorithm (WOA) optimized BiLSTM_Attention structure to solve this problem. The traffic flow is predicted first using the BiLSTM_Attention network which is then optimized by using the WOA to obtain its four best parameters, including the learning rate, the training times, and the numbers of the nodes of two hidden layers. Finally, the four best parameters are used to build a WOA_BiLSTM_Attention model. The proposed model is compared with both conventional neural network model and neural network model optimized by the WOA. Based on the evaluation metrics of MAPE, RMSE, MAE, and R2, the WOA_BiLSTM_Attention model proposed in this article presents the best performance.

KEYWORDS

attention, BiLSTM, prediction, traffic flow, whale optimization algorithm

1 INTRODUCTION

1.1 Background

Traffic flow is an important indicator of urban development and operation status. Predicting, adjusting and controlling traffic flow is of great significance to urban management.[1] How to effectively predict traffic flow trends for early intervention is one of the key issues in decision-making for traffic management.[2] Traffic flow data collected at fixed observation points often include lots of attributes such as traffic speed, vehicle flow, passing duration, and specific road conditions. Due to the large number of vehicles and complex transport networks, the data are characterized by large size, high variation, and high dynamics. In particular, the recent development of smart cities[3] lead to more and more sensors that are installed on vehicles, enabling the communication between vehicles and peripheral equipment at any time, such as vehicle to vehicle (V2V)[4] communication, vehicle to infrastructure (V2I)[5] communication, and vehicle to everything (V2X) information exchanges. Hence, vehicles can send their position information at any time. This Internet of Vehicles technology is applied based on integrating technologies such as GPS navigation,[6,7] and mass of traffic flow data are generated.[8] The mass of traffic datasets lay the foundation for traffic flow prediction. Accurate traffic flow prediction also allows urban terminals to carry out better traffic planning and to judge the operation condition of traffic networks in advance.

Related work

In recent years, various modeling methods have been applied to build stable and accurate traffic flow prediction models. The Auto-Progressive Integrated Moving Average (ARIMA) model, the Gaussian process, and the Kalman filter constitute the major part of conventional methods. For example, Liu[9] applied the ARIMA model to study the rail traffic flow. Emami[10] applied the Kalman filter to filter traffic flow fluctuations and predict traffic flow. These conventional models usually adopt linear methods that are too simple to reflect thenon-linear factors involved in the process of short-term traffic flow prediction, resulting in low prediction accuracy.

Machine learning and artificial intelligence are the core of conventional traffic flow prediction algorithms. Such algorithms include artificial neural networks (ANN),[11] extreme learning machines (ELM), and support vector regression (SVR). For example, Zhang[12] used the ELM non-iterative algorithm to predict air traffic flow. Castro-Neto[13] used the SVR algorithm to predict the short-term traffic flow. These methods have great advantages in tackling non-linear problems, but they are unable to tackle temporal correlation and to process large-scale data with poor prediction. Deep learning models have been further introduced in the load prediction area, including deep neural network (DNN),[14] stacked AutoEncoder (SAE),[15] and convolutional neural network (CNN).[16] Although compared with shallow ANNs, these DNNs have higher load prediction accuracy, and they also require artificial extraction of temporal characteristics. If the temporal characteristics of traffic flow data are ignored, artificial extraction of characteristics will affect the continuity of load data, and thus reduce the prediction accuracy of the models.

1.2 Major contribution

To meet the requirement of high accuracy of traffic flow prediction an whale optimization algorithm (WOA) optimized BiLSTM_Attention ("BiL-STM_A" for short) model is proposed in this article. BiLSTM is a bidirectional LSTM network structure that can process temporal networks better than the unidirectional LSTM and can extract traffic flow data in two directions.[17] Attention is a weight mechanism used to capture different weights of hidden layers and further to overcome the long-term dependence of networks such as RNNs and RNN-based improved networks when the input time series is long. When the network framework is established, obtaining the best hyperparameters of the network structure is difficult. Therefore, the BiLSTM_A model is optimized using the WOA to get the best parameters, so as to build a WOA_BiLSTM_Attention ("WOA_BiLSTM_A" for short) model.

2 RELEVANT THEORIES

2.1 BiLSTM short-term traffic flow prediction based on improved WOA optimized attention mechanism

This article proposes a BiLSTM_A model optimized using WOA. When the frameworks of models are established, it is difficult for many of them to directly obtain the best hyper parameters for one time. Even if the frameworks are the same, the networks of different hyper parameters have greatly different accuracies. Therefore, to predict short-term traffic flow, a WOA_BiLSTM_A model is proposed based on the BiLSTM, the attention mechanism, and the WOA. The model is optimized by using the WOA to improve its hyper parameter adaptability. The experiment results show that the optimized network is much better than the comparison networks.

2.2 Whale optimization algorithm

The WOA is a new heuristic optimization algorithm that mimics the hunting behavior of whales. Whales use a special hunting method called bubble-net hunting strategy.[18] The bubble-net shown in Figure 1. The WOA involves the following three stages: encircling prey, bubble-net attacking, and search for prey.[19] The process is shown in Figure 2.

2.2.1 Encircling prey

This stage mainly mimics the behavior of whales encircling the prey during hunting. To describe the behavior, the following model is proposed:

$$\vec{D} = \left| \vec{C} \bullet \overrightarrow{X^*}(t) - \vec{X}(t) \right|$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A} \bullet \vec{D}$$

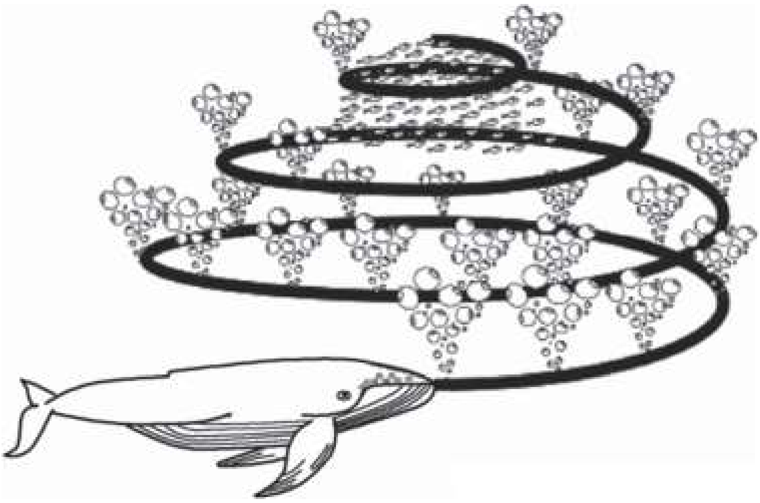


FIGURE 1 Whale hunting behavior diagram

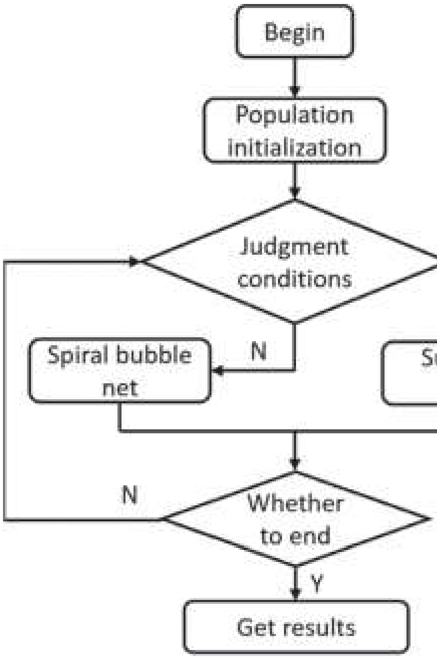


FIGURE 2 WOA algorithm flow chart

Where $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$, $\vec{C} = 2 \cdot \vec{a}_2$ and $\vec{a}$ decrease to 0 during the search, $\vec{a} = 2 - \frac{2t}{T_{max}}$ and T_{max} are the maximum number of iterations, $\vec{r}_1$ and $\vec{r}_2$ are random vectors meeting [0,1], t is the number of current iterations, $\overrightarrow{X^*(t)}$ represents the vector of the best whale position up to now, $\vec{X}(t)$ represents the vector of the current whale position, and $||$ represents the absolute value. If there is the best solution, X^* will be updated at each iteration.

2.2.2 Hunting behavior

Whales swim in a spiral way to prey, and the hunting behavior is expressed as follows:

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) + \overrightarrow{D'} \bullet e^{bl} \cos(2\pi l)$$

$\overrightarrow{D'} = |\overrightarrow{X^*}(t) - \overrightarrow{X}(t)|$ is the distance between a whale and the prey, b is a constant defining the shape of the logarithmic spiral, and l is the uniformly distributed random vector [−1,1]. Whales swim around the prey within a shrinking circle and along a spiral-shaped path simultaneously. So, we assume the probability of P_i to select the shrinking encircling method and the probability of $1 - P_i$ to select the spiral model to update the position of whales. The mathematical model is described as follows:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A} \bullet \vec{D}, & if\, p < 0.5 \\ \overrightarrow{X^*}(t) + \overrightarrow{D'} \bullet e^{bl} \cos(2\pi l), & if\, p \geq 0.5 \end{cases}$$

When the value of A is in [−1,1], the new position of a whale can be defined anywhere between its current position and the prey's position. The

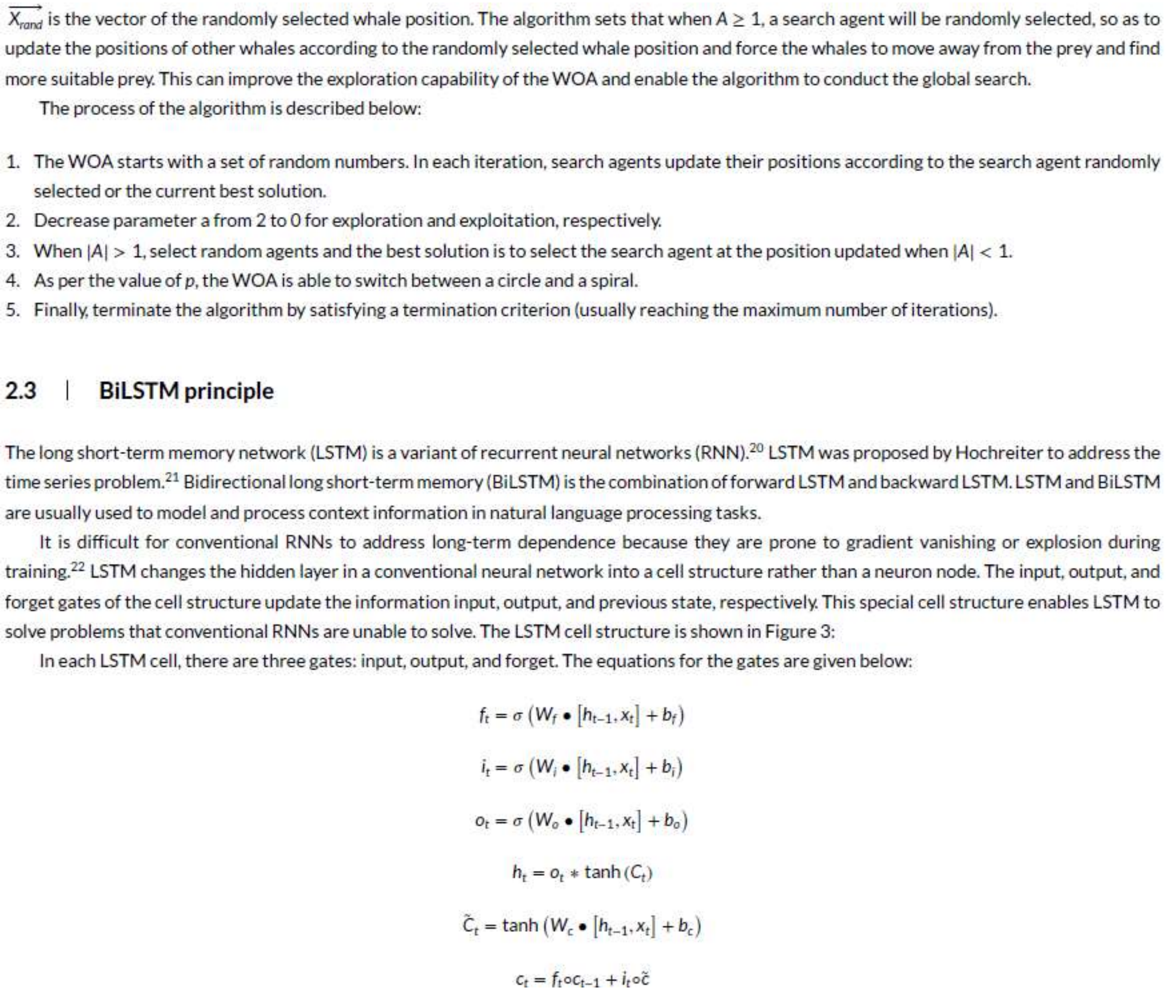
algorithm sets that when $A < 1$, the whales attack the prey.

2.2.3 Search for prey

The mathematical model for this phase is as follows:

$$\vec{D} = \left|\vec{C} \bullet \overrightarrow{X_{rand}} - \vec{X}(t)\right|$$

$$\vec{X}(t-1) = \overrightarrow{X_{rand}} - \vec{A} \bullet \vec{D}$$



$\overrightarrow{X_{rand}}$ is the vector of the randomly selected whale position. The algorithm sets that when $A \geq 1$, a search agent will be randomly selected, so as to update the positions of other whales according to the randomly selected whale position and force the whales to move away from the prey and find more suitable prey. This can improve the exploration capability of the WOA and enable the algorithm to conduct the global search.

The process of the algorithm is described below:

1. The WOA starts with a set of random numbers. In each iteration, search agents update their positions according to the search agent randomly selected or the current best solution.
2. Decrease parameter a from 2 to 0 for exploration and exploitation, respectively.
3. When $|A| > 1$, select random agents and the best solution is to select the search agent at the position updated when $|A| < 1$.
4. As per the value of p, the WOA is able to switch between a circle and a spiral.
5. Finally, terminate the algorithm by satisfying a termination criterion (usually reaching the maximum number of iterations).

2.3 | BiLSTM principle

The long short-term memory network (LSTM) is a variant of recurrent neural networks (RNN).[20] LSTM was proposed by Hochreiter to address the time series problem.[21] Bidirectional long short-term memory (BiLSTM) is the combination of forward LSTM and backward LSTM. LSTM and BiLSTM are usually used to model and process context information in natural language processing tasks.

It is difficult for conventional RNNs to address long-term dependence because they are prone to gradient vanishing or explosion during training.[22] LSTM changes the hidden layer in a conventional neural network into a cell structure rather than a neuron node. The input, output, and forget gates of the cell structure update the information input, output, and previous state, respectively. This special cell structure enables LSTM to solve problems that conventional RNNs are unable to solve. The LSTM cell structure is shown in Figure 3:

In each LSTM cell, there are three gates: input, output, and forget. The equations for the gates are given below:

$$f_t = \sigma\left(W_f \bullet \left[h_{t-1}, x_t\right] + b_f\right)$$

$$i_t = \sigma\left(W_i \bullet \left[h_{t-1}, x_t\right] + b_i\right)$$

$$o_t = \sigma\left(W_o \bullet \left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

$$\tilde{C}_t = \tanh\left(W_c \bullet \left[h_{t-1}, x_t\right] + b_c\right)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}$$

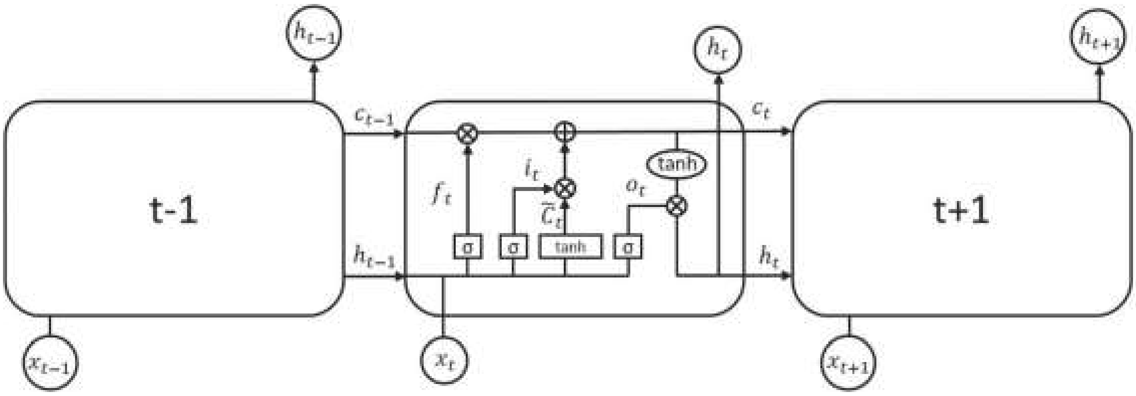


FIGURE 3 LSTM cell structure diagram

Where "∘" represents multiplying one element at the corresponding position of a vector by another element at the corresponding position of another vector, σ represents the sigmoid function used as the activation function, f represents the forget gate, i represents the input gate, o represents the output gate, C represents the cell state, $\breve{c}$ represents the unit state of the current input, and h represents the cell output.[23,24]

The BiLSTM is improved for bidirectional input over the LSTM.[25] It sets an input gate, a forget gate, and an output gate to solve the problem of long-term dependence absence of recurrent neural memory networks. Leveraging the bidirectional input, it can capture series features from both positive and negative directions and thus learn the features information from more perspectives. Bi-LSTM uses the bidirectional RNN input mode and replaces the recurrent unit in RNN with the LSTM recurrent unit that has a gate control unit, equivalent to building a unidirectional LSTM network at both ends of the series, and both the networks are connected to the same layer. This structure provides complete context information in the input series of the output layer and learns the series features from both positive and negative directions.

The positive direction of the neural network is updated as follows:

$$\overrightarrow{h_t} = H\left(W_{x\overrightarrow{h_t}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h_{t-1}} + b_{\overrightarrow{h}}\right)$$

The negative direction of the neural network is updated as follows:

$$\overleftarrow{h_t} = H\left(W_{x\overleftarrow{h_t}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h_{t+1}} + b_{\overleftarrow{h}}\right)$$

The combination output of the bidirectional recurrent neural network layer is as follows:

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h_y} + W_{\overleftarrow{h}y} \overleftarrow{h_t} + b_y$$

Where t is the time series, h is the hidden layer vector corresponding to the subscript time, x is the input corresponding to the subscript time, y is the output corresponding to the subscript time, W is the weight matrix between the corresponding subscript input and hidden layer, between hidden layers, and between hidden layer and output, b is the offset vector of the corresponding subscript hidden layer or output layer, and H is the sigmoid activation function of the hidden layer.[26]

2.3 Attention mechanism

The attention mechanism is a solution to mimicking human attention and a means of allocating attentional resources. In some cases, people concentrate on what is worthy of attention at certain moments. In this process, they often ignore other areas so as to obtain more details worthy of attention and suppress other useless information. The core principle of this algorithm is how to rationally and skillfully change the attention to the information concerned, ignore irrelevant information, and amplify the necessary information to the maximum extent.[27] The attention structure is shown in Figure 4. Where $x_t(t \in [1, n])$ represents the input of the network, $h_t(t \in [1, n])$ corresponds to the output of the hidden layer obtained by the input layer, $\alpha_t(t \in [1, n])$ is the attention probability distribution value output by the attention mechanism to the hidden layer, and y is the output value of the network introduced by the attention mechanism.

Conventional encoder-decoder RNN models often have a problem that the input series, regardless of their length, are all encoded to vector representation with a fixed length, while decoding is limited by the vector representation of that fixed length. This problem badly restricts the

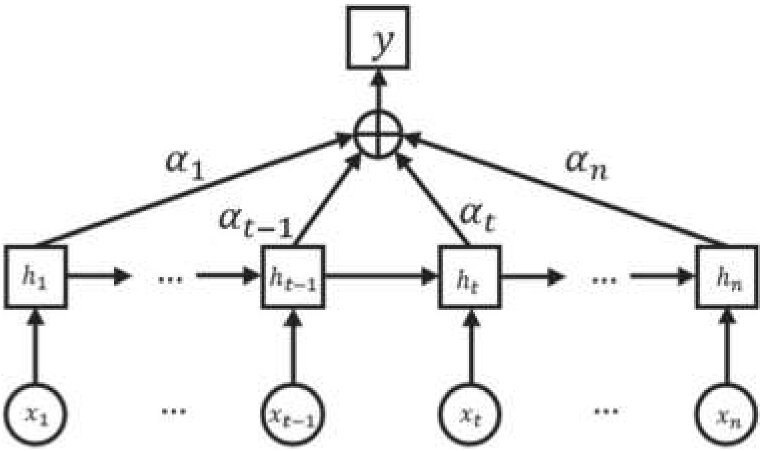


FIGURE 4 Attention structure diagram

performance of the models. Especially, when input series are long, their performance will be very poor. The attention mechanism breaks the limitation that conventional encoder-decoders rely on the internal fixed-length vector during encoding. They retain the intermediate output results of the input series from the encoders and then train a model to selectively learn these inputs and correlate the output series with them during the model's output. In other words, the probability of success of each item of the output series depends on which items are selected from the input series. This is the core principle of the attention mechanism.

3 ALGORITHM PROCESS

3.1 BiLSTM_A model

The attention mechanism attracted much attention when it was proposed. Like human beings attaching more importance to some information, the attention mechanism can properly assign weights to the information obtained and perform summation based on the weights.[28] As a result, the attention method is highly interpretable and more effective than other methods. In the early stage, researchers integrated the attention mechanism and the BiLSTM to address text translation and classification problems.[29] In this article, the BiLSTM is used to process traffic flow data, and essentially the time series. The network structure of the BiLSTM_A model is shown in Figure 5.

The BiLSTM_A structure is divided into the following five layers:

1. Input Layer: inputting series. The series may be character series or time series, or a combination of both. In this article, the input traffic flow is time series.
2. Embedding Layer: mapping each time series into a low-dimensional vector. The embedding layer of the model covers the embedding of time series words and the embedding of relative position codes. A vector may be randomly initialized or a trained vector may be used.
3. LSTM Layer: using the BiLSTM to obtain advanced features from the previous step.
4. Attention Layer: generating a weight vector and multiplying it by the weight vector to combine the short time series of each time step into a long time series feature vector.
5. Output Layer: finally, outputting by using the time series.

As shown in Figure 5, the main difference between the previous conventional BiLSTM models and the BiLSTM_A model is that in the latter model, a structure called Attention Layer is interposed after the BiLSTM layer and before the full connection to the softmax layer. The Attention structure first calculates the weight of the time series of each position in the BiLSTM outputs, and then performs weighted summation and uses the sum as the representation vector of the time series, and finally conducts output and prediction.

The calculation formula of Attention is as follows:

$$M = \tanh(H)$$

$$\alpha = \mathrm{softmax}\left(\omega^T M\right)$$

$$r = H\alpha^T$$

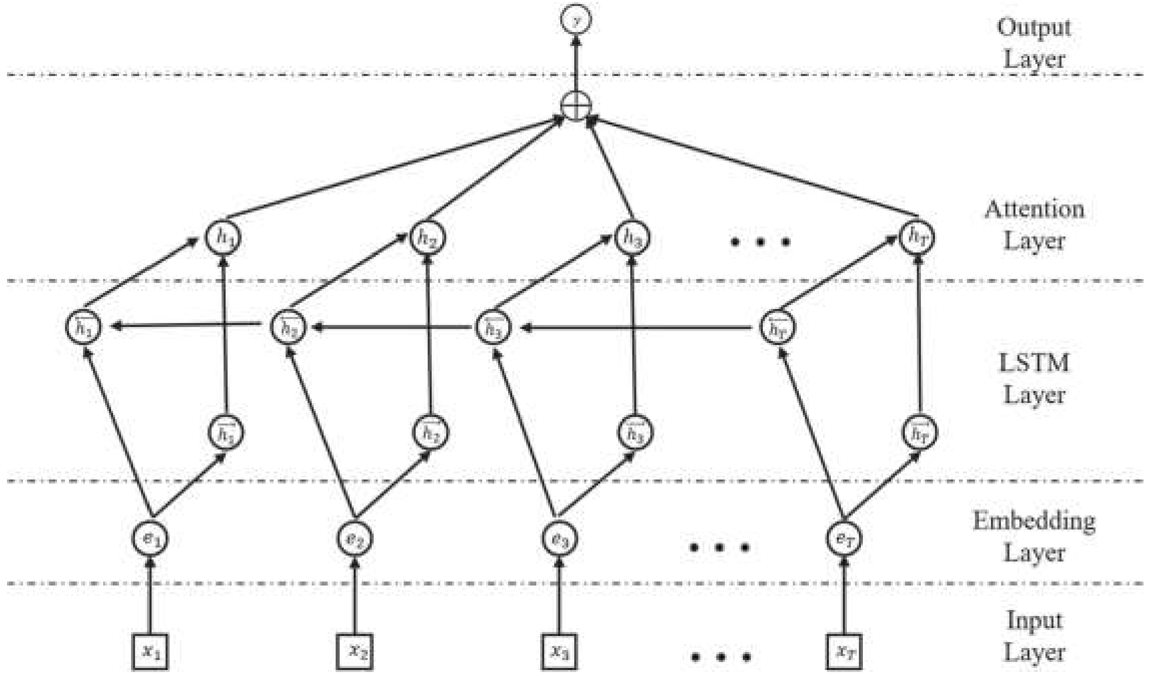


FIGURE 5 The network structure of BiLSTM_A model

Where T is the length of the input time series, and the weighted sum of these output vectors constitutes the time series r. Where $\in \mathrm{R}^{d^{\omega} \times T}$, d^{ω} is the dimension of the time series vector. ω is a trained parameter vector, ω^{T} is the transposed vector, and dimensions of ω, α, r are $d^{\omega}, T, d^{\omega}$, respectively.

Obtain the final time series representation from the following formula:

$$h^{*} = \tanh(r)$$

Finally, we use the softmax classifier to predict$\hat{y}$, the label of a discrete class Y set. The classifier uses the hidden * as the input.

$$\hat{p} = (y|S) = softmax\left(W^{(s)}h^{*} + b^{(s)}\right)$$

$$\hat{y} = \underset{y}{\mathrm{argmax}}(y|S)$$

The loss function is the negative log-likelihood value of the real label$\hat{y}$.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} t_i \log(y_i) + \lambda\|\theta\|_F^2$$

Where λ is the regularization parameter of L2. In this article, we alleviate overfitting also through the regularization of dropout and L2.

3.2 Hyperparameters of WOA optimized BiLSTM_A

The WOA effectively eliminates the defect that the BiLSTM_A algorithm is prone to the local best solution and improves the accuracy of parameter optimization.

The WOA optimization is to obtain the maximum or minimum value of the fitness function. In this article, the mean square error of the minimum BiLSTM_A network output and the actual value is used as the fitness function. The formula is given below:

$$\mathrm{TrainingLoss} = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}$$

Where y_i is the ith actual value in the prediction result, $\hat{y}_i$ is the ith predicted value in the prediction result, and N is the total number of samples. The more accurate the predicted values, the smaller the loss values obtained.

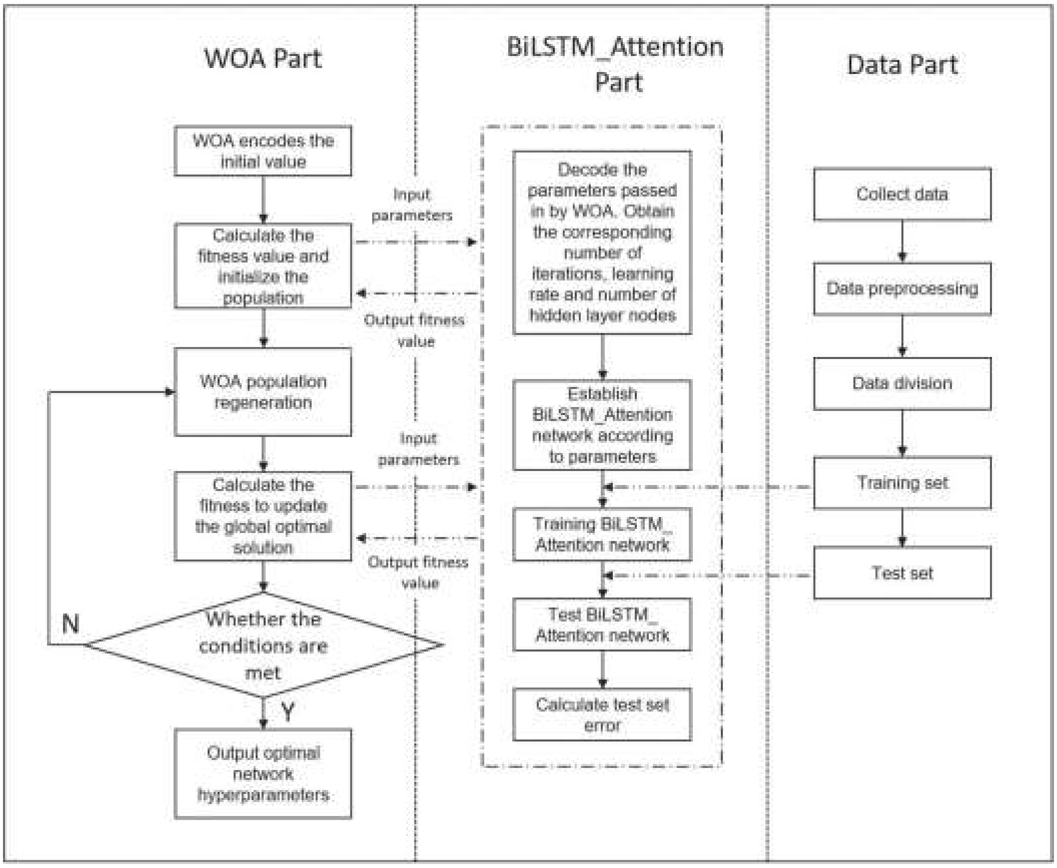

WOA Part
BiLSTM_Attention Part
Data Part
WOA encodes the initial value
Calculate the fitness value and initialize the population
WOA population regeneration
Calculate the fitness to update the global optimal solution
Whether the conditions are met
N
Y
Output optimal network hyperparameters
Input parameters
Output fitness value
Input parameters
Output fitness value
Decode the parameters passed in by WOA. Obtain the corresponding number of iterations, learning rate and number of hidden layer nodes
Establish BiLSTM_Attention network according to parameters
Training BiLSTM_ Attention network
Test BiLSTM_ Attention network
Calculate test set error
Collect data
Data preprocessing
Data division
Training set
Test set

FIGURE 6 WOA_ BiLSTM_ A algorithm flow chart

The flow chart of the prediction model of the WOA_BiLSTM_A short-term traffic flow algorithm is shown in Figure 6.

According to Figure 6, the WOA optimized BiLSTM_A algorithm includes three parts: WOA, BiLSTM_A, and data. The WOA part is the detailed flow of whale algorithm. The BiLSTM_A part implements the detailed prediction algorithm. The data part is used to process data. Each part transfers parameters during model training. The prediction process of the model is described below:

1. Initialize BiLSTM_A model parameters.
2. Initialize WOA algorithm population. Build a set of values with the four variables (iter, α, n_1, n_2) and puts them as optimization parameters into the WOA. The four parameters represent the number of iterations, the learning rate, and the number of the nodes of the two hidden layers of the BiLSTM network.
3. Assign the initialized values as the historical best value to the parameters of BiLSTM_A and train the model. Then, predict the test set and output the mean square error of the actual output value and the expected output value as the *TrainingLoss*.
4. Set the *TrainingLoss* obtained from the conventional BiLSTM_A training as the system requirement. Adjust the parameters for the WOA according to the fitness to update the best solution of the population, and obtain the loss value of the WOA optimized model.
5. If the parameters of the model optimized by the WOA are less than the *Trainingloss*, which means that the requirement is met, output the final prediction model and parameter values.
6. If the loss value cannot be less than the *TrainingLoss* or the number of iterations does not reach the maximum, update the parameters and carry out training again. Otherwise, stop training.

4 PREDICTION BASED ON THE EXPERIMENT OF WOA OPTIMIZED BILSTM_A

4.1 Data processing and experiment conditions

The development of smart cities drives the acquisition of data including lane inspection and tracking as well as the location, speed and direction of vehicles from theory to reality.[30] Acquisition of mass of traffic flow data was theoretical in the past, and now this is possible.[31] In this context, the set

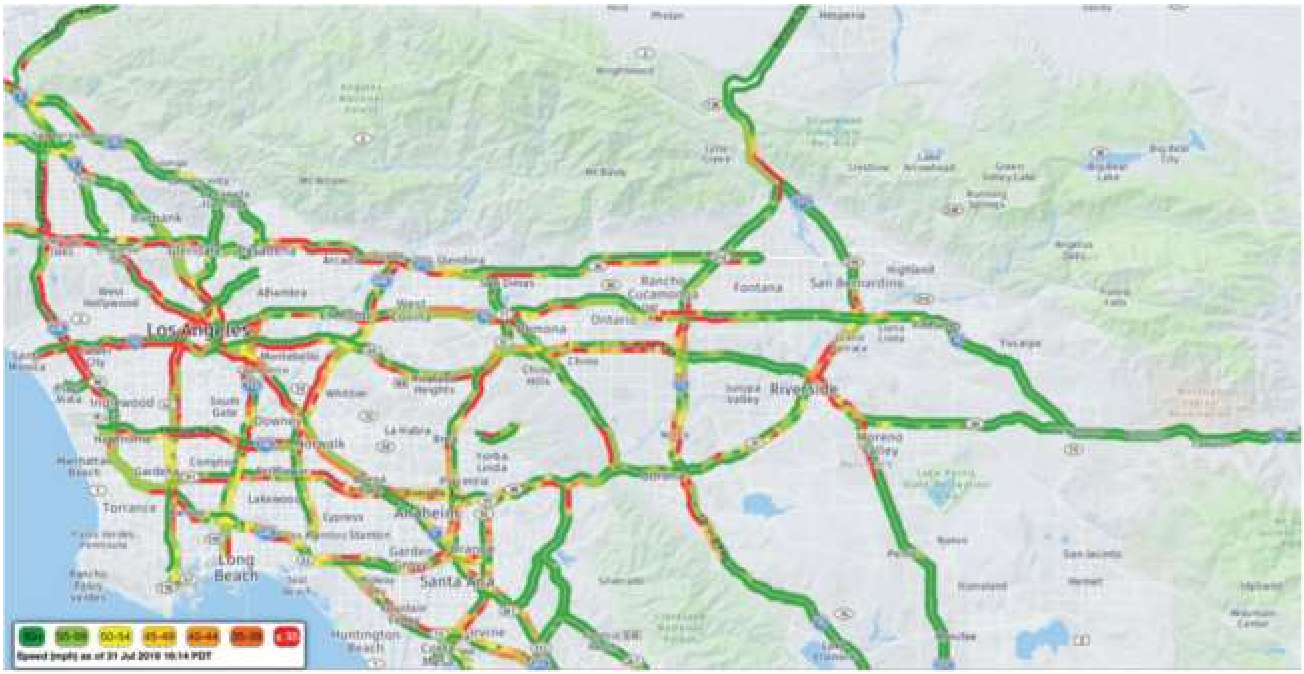


FIGURE 7 Range of the data

TABLE 1 Data division table

Classification	Input dimension size	Output dimension size
Training set	(47,832,24)	(47,832,1)
Test set	(144,24)	(144,1)

of traffic flow data of California State Route 24 are used in this article. Refer to Figure 7 for the range of the data. The data were sampled every 5 min. The data from 2000 time intervals after 00: 00: 00 on May 1, 2014 are selected for the experiment in this article. There are 2000 × 24 = 48,000 values in total.

4.2 Experiment conditions and parameters

The python3.7 is used in the experiment. The hardware platform is Intel (R) Core (TM) i9-10900X CPU @3.70GHZ, 64 GB memory, 1 TB solid-state SSD, NVIDIA GeForce RTX 2080Ti graphics card. TensorFlow 2.2 and Tensor flow 1.14 are used to train and test the model.[32] Because the data are time series, a rolling series model is built. To be specific, the values of data 1 to n are the input, the value of data $n + 1$ is the tag output; the values of data 2 to $n + 1$ are the input, and the value at the time $n + 2$ is the tag output, and so on. To facilitate the training of the model network, the original data are standardized with the StandardScaler method. StandardScaler makes the processed data conform to the standard normal distribution.[33] Namely, the mean is 0, the standard deviation is 1, and the conversion function is as follows:

$$x^{*} = \frac{x - \mu}{\sigma}$$

Where μ is the mean of all the sample data and σ is the standard deviation of all the sample data.

Here, n = 24. Classify the training set and test set for the original data. The size of data classified is shown in Table 1.

4.3 Comparison of models

To measure the optimization effect of the algorithm proposed in this article more objectively, an experiment is carried out by comparing the WOA_BiLSTM_A network with the BP neural network, LSTM neural network, CNN_LSTM_Attention network (CLSTM_A network), WOA optimized

CLSTM_A network (W_CLSTM_A network), and BiLSTM_A network. All the hyperparameters of the BP network, LSTM network, CLSTM_A network, and BiLSTM_A network are set to the same, so as to reasonably analyze the influence of each model hyperparameter on the network structure. They are trained to converge during network training. The parameters of the four comparison models are shown in Table 2.

As shown in Table 2 above, the hyperparameters of network nodes in the first and second layers of the four models are all 100, and the first layer of the CLSTM_A network is a one-dimensional convolution layer.

Refer to Figure 8 for the loss function curves after the simulation of the four networks.

As shown in Figure 8, the BP neural network converges after three rounds of training. The loss value of the LSTM neural network quickly drops before the 20th round of training and slowly drops after the 20th round until both the training set and verification set converge. The loss value of the CLSTM_A network quickly drops before the 50th round of training and gradually converges after the 50th round of training. The loss value of the BiLSTM_A network quickly drops before the 75th round and gradually converges after the 75th round.

TABLE 2 Comparison model parameter table

	Parameter					
Network type	Number of nodes in the first layer	Number of nodes in the second layer	Loss function	Optimizer	Number of iterations	Batch_size
BP	100	100	mse	adam	100	128
LSTM					100	
CLSTM_A					200	
BiLSTM_A					200	

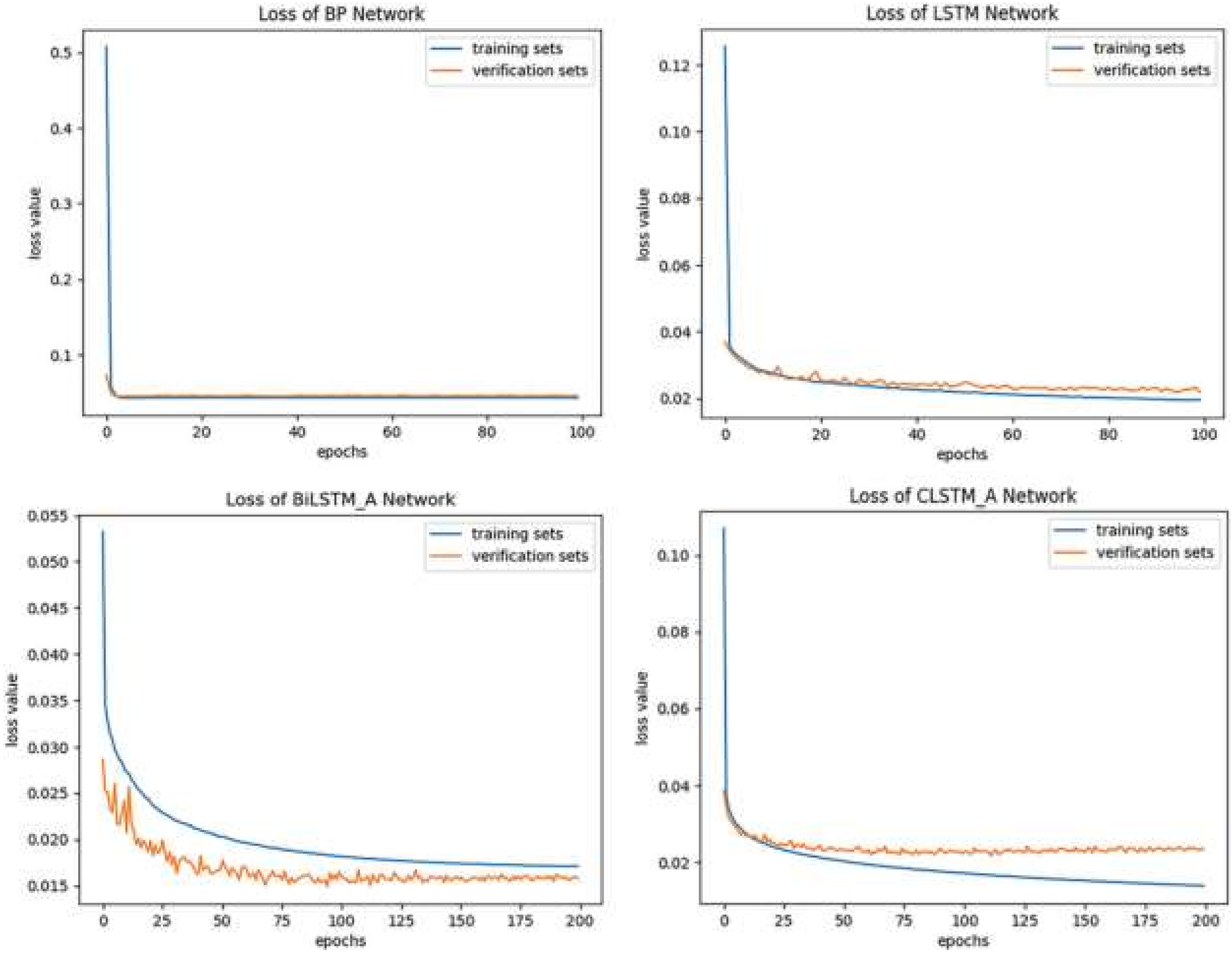


FIGURE 8 Compare network loss function curve

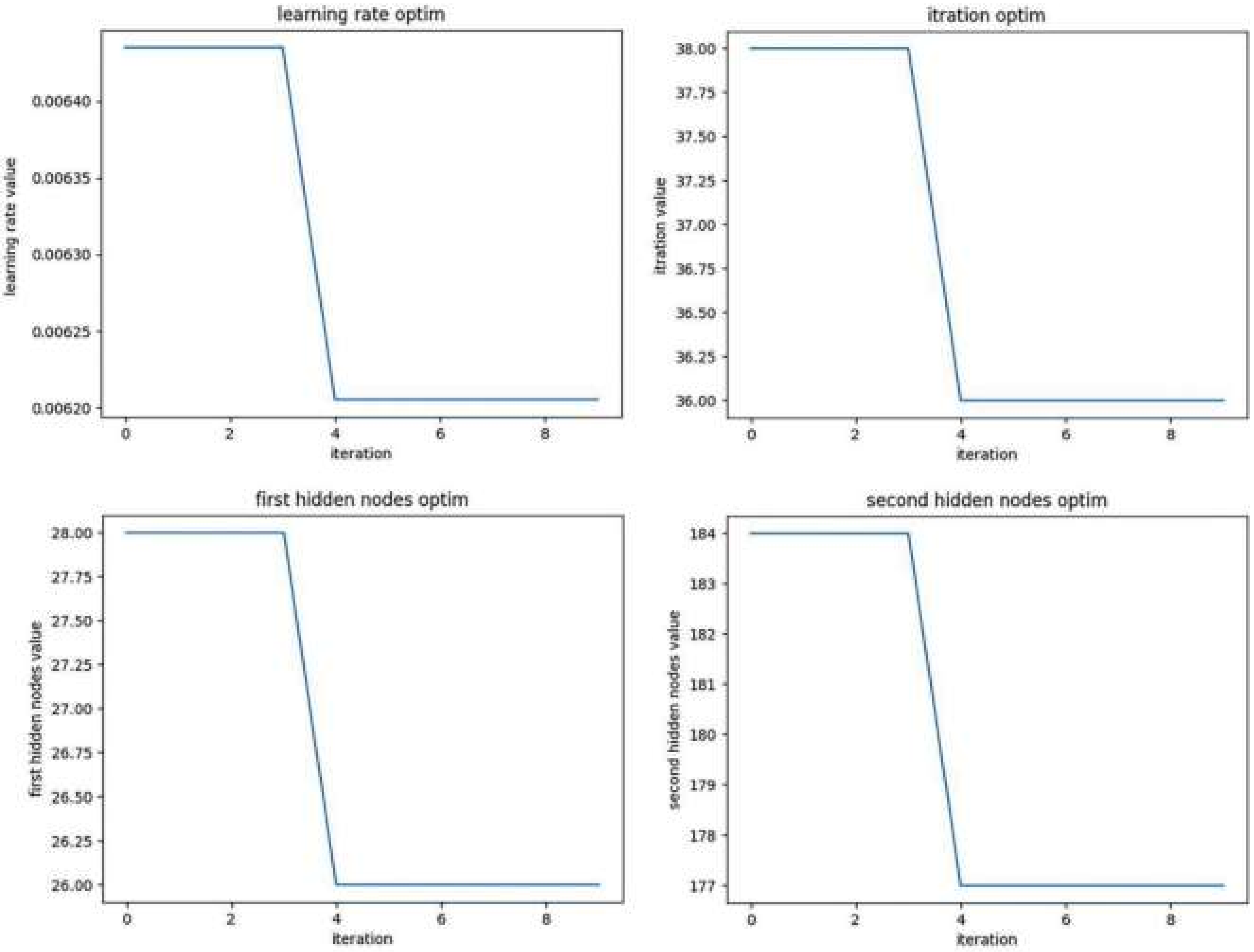


FIGURE 9 WOA optimized CLSTM_A fitness curve

In order to verify the effectiveness of WOA algorithm, as shown by the comparison networks above, this article also optimizes the CLSTM_A model with the WOA and uses it as the comparison model of WOA_BiLSTM_A.[34] The number of the initial populations of the WOA is set to 5, the number of iterations is set to 10, and the dimensions are set to 4. As for the learning rate and the numbers of the nodes of the two hidden layers of the CLSTM_A and the BiLSTM_A, we set the optimization lower bound to [0.001,10,1,1] and the optimization upper bound to [0.01,200,200,200].

The WOA is used for parameter optimization to obtain the best learning rate, training times, and numbers of the nodes of the two hidden layers of the CLSTM_A and BiLSTM_A models optimized by WOA. After simulation calculation, the fitness convergence curves of the WOA in the process of optimizing the two models are obtained. Refer to Figures 9 and 10.

According to Figures 9 and 10, the CLSTM_A model has the best fitness value at the fourth iteration, and the corresponding minimum mean square error is 0.0139; the BiLSTM_A model has the best fitness value at the second iteration and the corresponding minimum mean square error is 0.0134. The best parameter combinations of the CLSTM_A and the BiLSTM_A optimized by WOA are given in Table 3.

The best parameter combinations of the optimized CLSTM_A and BiLSTM_A are used for simulation calculation and the loss function curves are shown in Figure 11.

According to Figure 11, in the case of the best parameter combinations, the loss functions can finally converge in a certain range after a quick drop in the early stage, demonstrating the effectiveness of the simulation.

4.4 Comparison of experiment results

In this article, it is known that the best learning rate of the WOA_BiLSTM_A model is 0.00687, its number of iterations is 78, its number of the nodes of the first layer is 62 and that of the second layer is 191. As per the classification of the training set and the test set, the BP neural network,

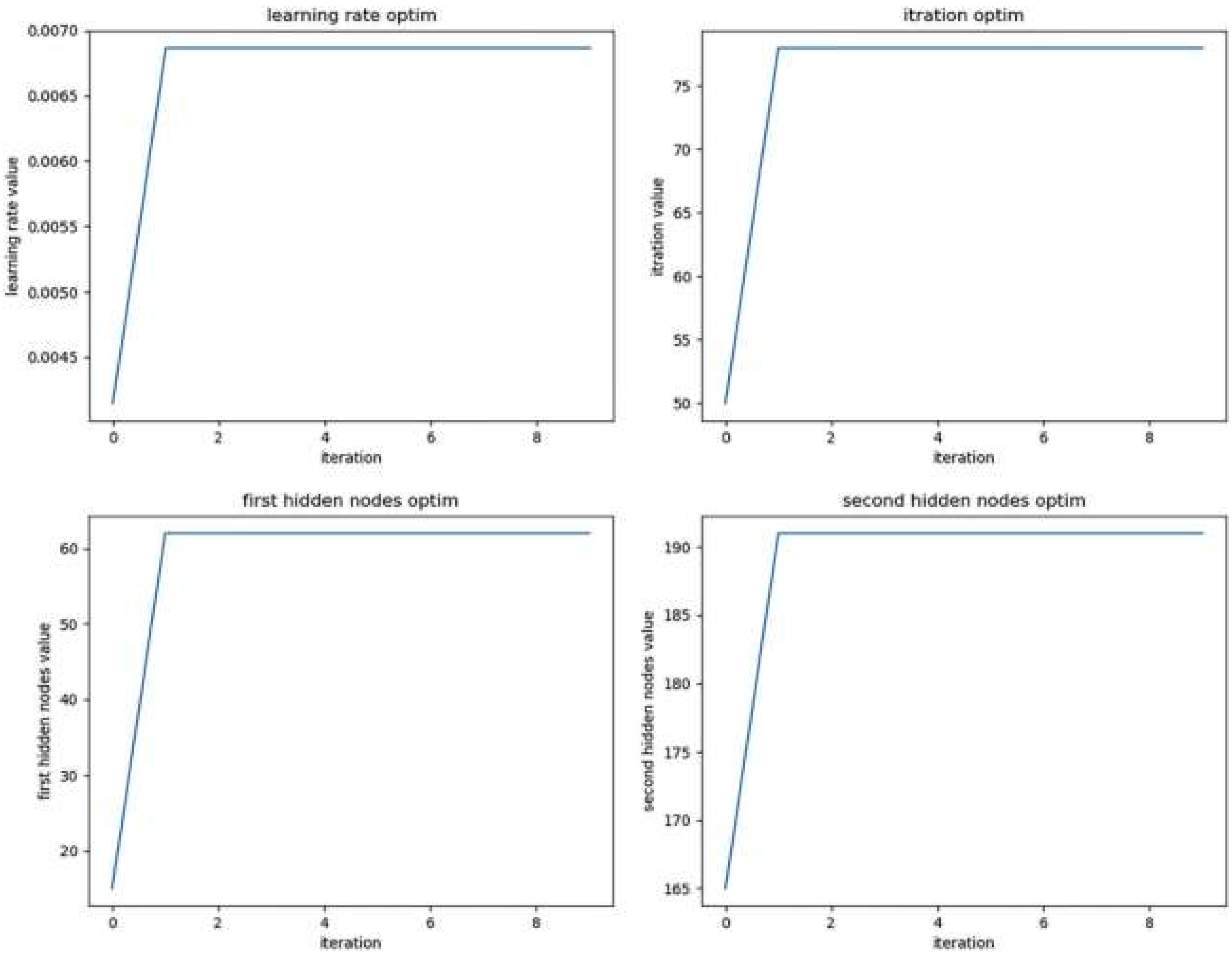


FIGURE 10 WOA optimized BiLSTM_A fitness curve

TABLE 3 Optimal parameter combination of two models after WOA optimization

Network type	Optimal learning rate	Number of iterations	Number of nodes in the first layer	Number of nodes in the second layer
W_CLSTM_A	0.00621	36	26	177
WOA_BiLSTM_A	0.00687	78	62	191

LSTM neural network, CLSTM_A network, W_CLSTM_A network, and BiLSTM_A network are used as comparison networks of WOA_BiLSTM_A in the experiment. MAPE, RMSE, and MAE[35,36] of predicted and actual values and the linear regression coefficient of determination R2 are used to evaluate the errors, so as to better reflect the error distance between the predicted values and the actual values of different models. The MAPE, RMSE, and MAE formulas are given below. Where the predicted value is assumed to be $\hat{y} = \{\hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_n\}$ and the actual value is assumed to be $y = \{y_1, y_2, y_3, \ldots, y_n\}$.

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|$$

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

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

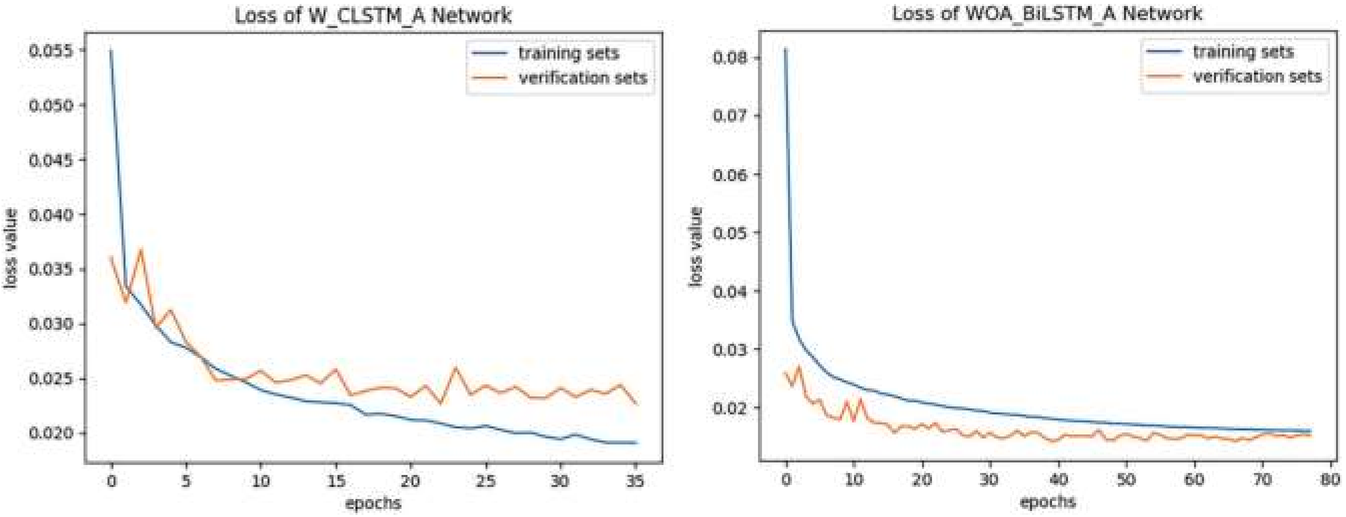


FIGURE 11 CLSTM_A and BiLSTM_A optimal parameter combination loss function curve

TABLE 4 Error analysis table

	Evaluating indicator			
Network type	MAPE	RMSE	MAE	R2
BP	0.0623	26.5267	21.0652	0.9139
LSTM	0.0382	16.8932	12.7013	0.9621
BiLSTM_A	0.0378	17.4511	12.9509	0.9638
CLSTM_A	0.0395	17.6930	12.7199	0.9617
W_CLSTM_A	0.0383	17.4843	13.2746	0.9626
WOA_BiLSTM_A	0.0361	17.1479	12.4787	0.9640

For MAPE, RMSE, and MAE, the smaller the values are, the more accurate the prediction results will be.

The linear regression coefficient R^2 is defined as follows:

$$R^2 = \frac{\mathrm{SSR}}{\mathrm{SST}} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

Where SSR is the sum of squares regression, SST is the sum of squares total, and $\overline{y}$ is the mean of actual values of y. The larger the value of R2 is, the more accurate the prediction results will be.

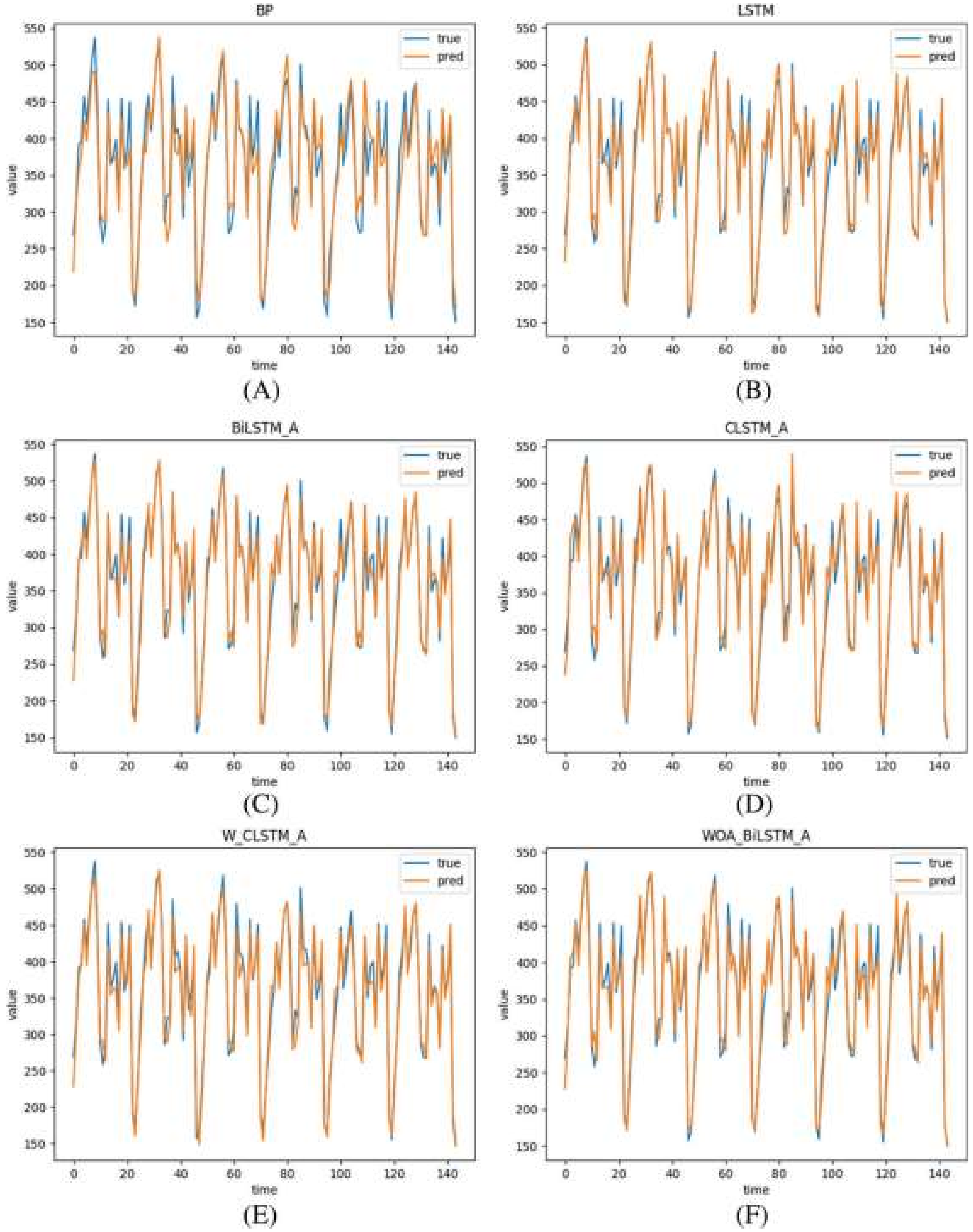
The comparison of error results from these evaluation metrics is shown in Table 4.

As shown in Table 4, the WOA_BiLSTM_A network is better than other networks in terms of each parameter evaluation metrics. Even after the WOA optimized CLSTM_A model is added in the comparison experiment, the WOA_BiLSTM_A is better than expected.

The trained model is used to test the test set. Figure 12 below shows the comparison of the relative actual values of the six models: BP, LSTM, BiLSTM_A, CLSTM_A, W_CLSTM_A, and WOA_BiLSTM_A. It can be seen from Figure 12 that the accuracy of the WOA_BiLSTM_A is slightly higher than that of other networks.

5 CONCLUSION

At present, deep learning technology is rapidly developing. Based on many scholars' researches on deep learning technology in the field of traffic flow prediction,[37] this article proposes a BiLSTM_A traffic flow prediction network model optimized using the WOA. The BiLSTM network is effective in extracting the time series feature. On this basis, the attention mechanism is used to capture different weights of the hidden layers of the BiLSTM



(A) (B) (C) (D) (E) (F)

FIGURE 12 Comparison between the predicted value of each network and the real value

network. When a network architecture is built, it is difficult to directly build a network of best hyperparameters for the data used at one time. Therefore, the WOA is used to optimize the parameters of the network structure and obtain the four parameters, that is, the best learning rate, training times, and number of the nodes of two hidden layers. Finally, the best parameters are imported into the BiLSTM_A network, and after training, the best traffic flow prediction network WOA_BiLSTM_A is built.

Dataset of Californian highway is used to train and validate the WOA_BiLSTM_A network; the BP, LSTM, CLSTM_A, BiLSTM_A, and W_CLSTM_A are used as the comparison models; MAPE, RMSE, MAE, and R2 are used as the valuation metrics. The result adequately proves that the network proposed in this article is much better than the comparison networks. Here, the BiLSTM neural network, the attention mechanism, and the WOA are integrated, and as a result, the accuracy of the short-term traffic flow prediction model is further improved. The significance of this article lies in that by optimizing the parameters with the WOA, the network accuracy can be greatly improved based on the best parameters of the network obtained by using the optimization algorithm in the case of different network frameworks.

To solve the problems arising in the application of the WOA_BiLSTM_A model to traffic flow prediction, future efforts can be made in the two aspects below:

1. In view of the high time complexity of the WOA in parameter optimization, the algorithm may be improved or other better algorithms may be used for parameter optimization to reduce the time complexity.
2. The network structure adopted in this article only uses nodes in two layers. Next, a deeper network structure may be used and more parameters may be optimized to improve the stability of the model in various cases.

ACKNOWLEDGMENTS

This research was supported by the National Key Research and Development Program of China (2019YFE0126100), the Key Research and Development Program in Zhejiang Province of China (2019C54005).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in california department of transportation at https://pems.dot.ca.gov/.

REFERENCES

1. Jin KH, Wi JA, Lee EJ, Kang SJ, Kim SK, Kim YB. TrafficBERT: pre-trained model with large-scale data for long-range traffic flow forecasting. *Expert Syst Appl*. 2021;186:115738.
2. Shahid N, Shah MA, Khan A, Maple C, Jeon G. Towards greener smart cities and road traffic forecasting using air pollution data. *Sustain Cities Soc*. 2021;72:103062.
3. Marques P. Uneven innovation: the work of smart cities. *Reg Stud*. 2021;55(8):1488-1489.
4. Hassan MU, Rehmani MH, Chen J. Privacy preservation in blockchain based IoT systems: integration issues, prospects, challenges, and future research directions. *Futur Gener Comput Syst*. 2019;97:512-529.
5. Hussein H, Radwan MH, Elsayed HA, Abd el-Kader SM. Depth-first-search-tree based D2D power allocation algorithms for V2I/V2V shared 5G network resources. *Wirel Netw*. 2021;27:1-15.
6. Zhao Y, Zhou X, Xu X, et al. A novel vehicle tracking ID switches algorithm for driving recording sensors. *Sensors*. 2020;20(13):3638.
7. Cui Z, Zhao Y, Cao Y, Cai X, Chen J. Malicious code detection under 5G HetNets based on multi-objective RBM model. *IEEE Netw*. 2021;35(2):82-87.
8. Cai X, Geng S, Wu D, Cai J, Chen J. A multi-cloud model based many-objective intelligent algorithm for efficient task scheduling in Internet of Things. *IEEE Internet Things J*. 2020;8(12):9645-9653.
9. Liu SY, Liu S, Tian Y, Sun QL, Tang YY. Research on forecast of rail traffic flow based on ARIMA model. *J Phys Conf Ser*. 2021;1792(1):012065.
10. Emami A, Sarvi M, Bagloee SA. Short-term traffic flow prediction based on faded memory Kalman filter fusing data from connected vehicles and Bluetooth sensors. *Simul Model Pract Theory*. 2020;102:102025.
11. Mabel MC, Fernandez E. Analysis of wind power generation and prediction using ANN: a case study. *Renew Energy*. 2008;33(5):986-992.
12. Zhang Z, Zhang A, Sun C, Xiang S, Guan J, Huang X. Research on air traffic flow forecast based on ELM non-iterative algorithm. *Mob Netw Appl*. 2021;26(1):425-439.
13. Castro-Neto M, Jeong YS, Jeong MK, Han LD. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Syst Appl*. 2009;36(3):6164-6173.
14. Li D, Li W, Wang X, Nguyen CT, Lu S. App trajectory recognition over encrypted internet traffic based on deep neural network. *Comput Netw*. 2020;179:107372.
15. Khodayar M, Kaynak O, Khodayar ME. Rough deep neural architecture for short-term wind speed forecasting. *IEEE Trans Industr Inform*. 2017;13(6):2770-2779.
16. Chen K, Chen K, Wang Q, He Z, Hu J, He J. Short-term load forecasting with deep residual networks. *IEEE Trans Smart Grid*. 2018;10(4):3943-3952.
17. Siami-Namini S, Tavakoli N, Namin A S. A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. 2019; arXiv preprint arXiv:1911.09512.
18. Jiang F, Wang L, Bai L. An improved whale algorithm and its application in truss optimization. *J Bionic Eng*. 2021;18(3):721-732.
19. Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Softw*. 2016;95:51-67.
20. Hu C, Duan Y, Liu S, et al. LSTM-RNN-based defect classification in honeycomb structures using infrared thermography. *Infrared Phys Technol*. 2019;102:103032.
21. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-1780.
22. Gao W, Gao J, Yang L, Wang M, Yao W. A novel modeling strategy of weighted mean temperature in China using RNN and LSTM. *Remote Sens*. 2021;13(15):3004.
23. Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. *Neural Comput*. 2000;12(10):2451-2471.

24. Kumar D, Mathur HD, Bhanot S, Bansal RC. Forecasting of solar and wind power using LSTM RNN for load frequency control in isolated microgrid. *Int J Model Simul*. 2021;41(4):311-323.
25. Jeong JH, Shim KH, Kim DJ, Lee SW. Brain-controlled robotic arm system based on multi-directional CNN-BiLSTM network using EEG signals. *IEEE Trans Neural Syst Rehabil Eng*. 2020;28(5):1226-1238.
26. Jia Y, Xu X. Chinese named entity recognition based on CNN-BiLSTM-CRF. Proceedings of the 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS); IEEE; 2018:1-4.
27. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. 2014; arXiv preprint arXiv:1409.0473.
28. Lan X, Wang H, Gong S, Zhu X. Deep reinforcement learning attention selection for person re-identification. 2017; arXiv preprint arXiv:1707.02785.
29. Zhou P, Shi W, Tian J, et al. Attention-based bidirectional long short-term memory networks for relation classification. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short papers); 2016:207-212.
30. Durduran SS. A decision making system to automatic recognize of traffic accidents on the basis of a GIS platform. *Expert Syst Appl*. 2010;37(12):7729-7736.
31. Hassan MU, Rehmani MH, Chen J. Differential privacy techniques for cyber physical systems: a survey. *IEEE Commun Surv Tutor*. 2019;22(1):746-789.
32. Qi L, Dou W, Chen J. Weighted principal component analysis-based service selection method for multimedia services in cloud. *Comput Secur*. 2016;98(1–2):195-214.
33. Spalink G, Freiburg V, Wagner P. Method for pre-processing digital data, digital to analog and analog to digital conversion system. August 4, 2005; U.S. Patent Application 11/034,584.
34. Gu B, Shen H, Lei X, Hu H, Liu X. Forecasting and uncertainty analysis of day-ahead photovoltaic power using a novel forecasting method. *Appl Energy*. 2021;299:117291.
35. Cui Z, Xu X, Xue F, et al. Personalized recommendation system based on collaborative filtering for IoT scenarios. *IEEE Trans Serv Comput*. 2020;13(4):685-695.
36. Li Y, Yu H, Song B, Chen J. Image encryption based on a single-round dictionary and chaotic sequences in cloud computing. *Concurrency Computat Pract Exper*. 2021;33(7):1-1.
37. Awan FM, Minerva R, Crespi N. Improving road traffic forecasting using air pollution and atmospheric data: experiments based on LSTM recurrent neural networks. *Sensors*. 2020;20(13):3749.

How to cite this article: Xu X, Liu C, Zhao Y, Lv X. Short-term traffic flow prediction based on whale optimization algorithm optimized BiLSTM_Attention. *Concurrency Computat Pract Exper*. 2022;34(10):e6782. doi: 10.1002/cpe.6782