Dynamic Optimization Long Short-Term Memory Model Based on Data Preprocessing for Short-Term Traffic Flow Prediction

YANG ZHANG\(^{1}\) AND DONGRONG XIN\(^{2}\)

\(^{1}\)School of Transportation, Fujian University of Technology, Fuzhou 350118, China
\(^{2}\)School of Civil Engineering, Fujian University of Technology, Fuzhou 350118, China

Corresponding author: Yang Zhang (zhang_yang1983@163.com)

This work was supported in part by the Natural Science Foundation of Fujian under Grant 2017H0003, in part by the Natural Science Foundation of Fujian under Grant 2019J01781, and in part by the Foundation of Fujian University of Technology under Grant GY-Z160123.

\section*{ABSTRACT}

In order to eliminate outliers in traffic flow data collection and promote the generalization performance of traffic flow prediction, this paper proposes a dynamic optimization long short-term memory (LSTM) model based on data preprocessing for short-term traffic flow prediction. A new classification algorithm named Asym-Gentle Adaboost with Cost-sensitive support vector machine (AGACS) is used for preprocessing traffic flow data. AGACS tries to employ Cost-sensitive SVM (CS-SVM) as weak component classifier in Asymmetric Gentle AdaBoost, and divide the data collection into outlier data and normal data. Only normal data is used for training LSTM to predict traffic flow and an improved chaotic Particle Swarm Optimization (CPSO) is used for dynamic optimizing hidden layer structure of LSTM to promote the generalization and robustness performance of model. The efficiency of the proposed method is proved with real traffic flow data, and the experimental results show that preprocess collecting data and dynamic optimize model structure are conducive to improve the performance of algorithm, and the proposed method achieved better performance than those of three other classical deep learning prediction methods.

\section*{INDEX TERMS}

Traffic flow prediction, deep learning, long short-term memory, support vector regression, particle swarm optimization.

\section*{I. INTRODUCTION}

Short-term traffic flow prediction is a hotspot issue in computer data processing and ITS. According to the traffic flow information about current and history data, make use of the computer data processing technology to analyze spatial-temporal evolution characteristics, short-term traffic flow prediction can scientific estimate the volume of traffic flow in the next 15 minutes. Accurate short-term traffic flow prediction not only can provide traffic information for travelers and assist them to formulate the reasonable travel path, but also the basic and necessary condition of administrator who can correctly implement traffic induction and control.

In order to achieve accurate short-term traffic flow prediction, many effective schemes are explored. Most of the early traditional short-term traffic flow prediction model based on statistical principle, and time-series models [1]–[3] are the most representative methods, which according to the continuity of traffic flow in temporal is derived. In this research field, the autoregressive integrated moving average model is the typical algorithm [4]. However, this kind of method lack of self-learning and adaptive ability. It is difficult to process the volatility data of short-term traffic flow, resulting to lower prediction accuracy.

Encourage from the excellent learning and data processing ability of machine learning methods, more and more short-term traffic flow prediction algorithms based on machine learning are put forward. Simultaneously, the prediction accuracy and robustness of models are improved obviously. There are some representative methods in this field such as support vector regression (SVR) [5]–[7], K-nearest neighbour (KNN) [8], artificial neural network (ANN) [9], [10]. Zhang et al. put forward a hybrid forecasting framework based on an improved SVR for traffic flow prediction. Cai et al. [12] proposed a hybrid traffic flow forecasting model combining gravitational search algorithm (GSA) and the SVR model. Marcin et al. [13] presented a data segmentation method, which was intended to
improve the performance of the KNN algorithm for making short-term traffic volume predictions. Cai et al. [14] proposed an improved KNN model to enhance forecasting accuracy based on spatiotemporal correlation and to achieve short-term traffic multistep forecasting. There are some ANN models developed for traffic flow forecasts, such as Tang et al. [15] proposed a hybrid method combing clustering methods and spatiotemporal correlation to predict future traffic trends based on ANN. Vlahogianni et al. [16] combined ANN and genetic algorithm to predict short-term traffic flow. These types of methods can find out inherent law through self-learning from large number of the traffic flow history data, which can enhance traffic flow prediction accuracy. Unfortunately, large number of accurate historical traffic flow data is not easy to get, hence the utility of methods are limited by this factor.

In order to overcome the problem of excessive dependence on the number of samples, and complete short-term traffic flow prediction under the condition of “small sample” to improve the practicability of algorithms, deep learning traffic flow prediction method make use of this field and gradually become the promising research field. The representative models in this field are long short-term memory neural network (LSTM) [17]–[19], gated recurrent unit neural network (GRU) [20]–[22], convolutional neural networks (CNN) [23]–[25], and deep belief network neural network model (DBN) [26]–[28]. Mou et al. [29] explored a temporal information enhancing LSTM (T-LSTM) to predict traffic flow of a single road section, which could improve prediction accuracy by capturing the intrinsic correlation between traffic flow and temporal information. Huang et al. [30] defined a deep architecture for traffic flow prediction, which learns features with DBN. Wu et al. [31] proposed a DNN based traffic flow prediction model (DNN-BTF) to improve the prediction accuracy, which CNN was also used to mine the spatial features and the GRU to mine the temporal features of traffic flow. Deep learning traffic flow prediction algorithms have stronger ability of self-learning data processing, which can achieve accurate prediction under the condition of a few typical true traffic flow historical data, and significantly improve the practicability of short-term traffic flow prediction methods.

However, there are still some problems in this kind of methods need to be solved. Firstly, thanks to extensive use of intelligent vehicle detection technology such as induction coil, video detection and laser radar, traffic flow information can be automatically captured using these technologies. Unfortunately, outliers are inevitable in the process of traffic flow data collection. They are less creditable and significantly different from the real data. Outliers lead to wrong spatial-temporal evolution rule of traffic flow in training process of deep learning, and that has very negative impact on prediction accuracy. In addition, the performances of these algorithms are greatly influenced by the model structure. Face with the diversity of traffic flow characters between different time periods, fixed model structure only adapt to a certain period of time in a day. How to structure applicable deep learning prediction model which can adapt the change of traffic flow in all-day is still in the air.

For solving these problems, a novel short-term traffic flow prediction method based on optimization combination LSTM and data preprocessing is proposed in this paper. There are two innovative points in the paper. Firstly, a new cost-sensitive learning algorithm is proposed, which is used to divide collecting data into two categories, outlier data and normal data. Outlier data is eliminated through the data preprocessing, and the normal data is retained as the training database of the model. Secondly, consider the divergence between the traffic flow characteristics at different time periods, such as weekday daytime, weekday nighttime, weekend daytime and weekend nighttime. The traditional LSTM model is unqualified for all time periods short-term traffic flow prediction. Therefore, we propose an improve LSTM model which can dynamic change and optimize the structure based on chaotic particle swarm optimization algorithm to adapt the change of the traffic flow characteristics at different time periods.

The rest of this paper is organized as follows: Section 2 presents the classification algorithm of Asym-Gentle Adaboost with Cost-sensitive SVM. Dynamic optimize structure of LSTM is introduced in Section 3. In Section 4, dynamic optimization AGACS-LSTM algorithm is proposed for short-term traffic flow prediction. Section 5 gives the experiments and analysis of results. Finally, the conclusion and future work are provided in Section 6.

II. THE CLASSIFICATION ALGORITHM OF ASYM-GENTLE ADABOOST WITH COST-SENSITIVE SVM

Outliers are inevitable in the process of traffic flow data collection. They are less creditable and significantly different from the real data [32]. Before traffic prediction, the data should be classified as real data and outlier data, and the outliers must be removed accurately. The proportion of outliers is smaller, which means sample data is positive and negative unbalance. This paper presents Asym-Gentle Adaboost with Cost-sensitive SVM (AGACS) classification algorithm. Considering cost-sensitive learning classifiers pay more attention to outliers and remove them accurately.

A. CS-SVM WEAK CLASSIFIER

The essential of cost-sensitive learning is that misclassification costs are designed according to sample sizes. Therefore, classifier is misclassification cost-sensitive. And it is very suitable to positive and negative unbalance sample data. Considering that samples of traffic flow are unbalance and difficult to be classified, in this paper, we introduce cost-sensitive learning into classification algorithm, and use CS-SVM as weak classifier.

Suppose a training sample set \( \{(x_i, y_i), i = 1, \ldots, n\} \) with \( n \) samples, where, \( x_i \in \mathbb{R}^d \) is sample characteristic space, \( y_i \in \{1, -1\} \) is category label, \( d \) is dimension of sample
space. In order to find the hyper-plane of optimal classifier with CS-SVM, the following optimization problems need to be solved:

\[
\min_{w, \xi, \rho} \frac{1}{2} \|w\|^2 - \lambda \frac{y}{N} \sum_{i \in I_+} \xi_i + \frac{1 - \lambda}{N} \sum_{i \in I_-} \xi_i \\
\text{s.t. } y_i (w \cdot \phi(x_i) + b) \geq p - \xi_i, \quad i = 1, 2, \ldots, N, \quad \xi_i \geq 0, \quad p \geq 0
\]

where, \(\omega\) and \(x_i\) are \(d\) dimensional column vectors, \(\gamma \in [0, 1]\) is the misclassification weight of cost-sensitive algorithm, which controls the cost sensitivity of classifier and the punishment value of misclassification, \(\nu \in [0, 1]\) is the parameter controlling support vector, \(\xi\) is slack variable, \(I_+ = \{i: y_i = +1\}\) and \(I_- = \{i: y_i = -1\}\) are sizes of positive and negative training samples, respectively.

The optimization problem can also be solved with its dual form by introducing Lagrange multiplier and negative training samples, respectively.

\[
\min_{\alpha} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \sum_i \alpha_i \\
\text{s.t. } 0 \leq \alpha_i \leq \frac{1}{\nu} y_i y_j, \quad \forall i, \sum_i y_i \alpha_i = 0, \quad \sum_i \alpha_i \geq \nu
\]

where, \(\alpha = (\alpha_1, \ldots, \alpha_N), 1\{A\} = 1\) if \(A\) is satisfied. Otherwise, it is 0.

Choose optimal \(\alpha^*(\gamma) = (\alpha_1^*(\gamma), \alpha_2^*(\gamma), \ldots, \alpha_N^*(\gamma))\) and solve the minimization problem in formula 3. Then the discriminant function of CS-SVM is expressed as:

\[
f(x) = \text{sign} \left( \sum_i \alpha_i^*(\gamma) y_i k(x_i, x) + b \right)
\]

where, \(k(x_i, x)\) is kernel function. In this paper, radial basis function \(k(x, y) = \exp(-||x-y||^2/2\sigma^2)\) is chosen as the kernel function of classifier.

### B. ASYMMETRIC GENTLE ADABOOST ALGORITHM

In this section, a new combinatorial algorithm of Asymmetric Gentle AdaBoost strong classifier is used for eliminating outlier data, which can enhance cost sensitivity to process the positive and negative unbalance problem, and CS-SVM is used in Asymmetric Gentle AdaBoost as weak classifier. This algorithm pays more attention to negative samples which are few numbers, and classifies them more accurately.

The core of Asymmetric Gentle AdaBoost algorithm is as follows. First, set the value of unbalanced cost loss function. \(C_1\) is the value of cost loss function, when positive samples are misclassified. \(C_2\) is the value of cost loss function, when negative samples are misclassified. The expression is this:

\[
\text{Loss}_{\text{asy}}(x_i, y_i) = \begin{cases} 
C_1 & y_i = 1 \cap H(x_i) = -1 \\
C_2 & y_i = -1 \cap H(x_i) = 1 \\
0 & \text{otherwise}
\end{cases}
\]

where, \(H(x_i) = \text{sign}(\sum j f_j(x_i))\) is the classification result of strong classifier, \(C_1, C_2 \in [0, 1]\) and \(C_1 > C_2\).

If number of misclassified positive samples is \(N_{FN}\), and number of misclassified negative samples is \(N_{FP}\), then the error rate of classifier is:

\[
\varepsilon_{\text{asy}}(C_1, C_2) = C_1 N_{FN} + C_2 N_{FP} = \sum_i A_{\text{loss}}(i)
\]

The upper limit of cost loss function expressed with exponential form is:

\[
A_{\text{loss}}(F(x)) = \begin{cases} 
e^{-C_1 y_i F(x_i)} & y_i = 1 \\
e^{-C_2 y_i F(x_i)} & y_i = -1
\end{cases}
\]

where, \(F(x)\) is an additive model, and its expression is \(F(x) = \sum_i f_i(x)\).

Therefore, the minimum of first round loss function with Asym-Gentle AdaBoost algorithm is:

\[
J_{\text{asy}}(F(x)) = E[I(y = 1)e^{-C_1 y F(x)} + I(y = -1)e^{-C_2 y F(x)}]
\]

where, \(I(.)\) is indicator function.

Sum of optimal weak classifiers in every round is \(F(x)\) in equation (8). It means that loss function in last round adds current optimal weak classifier \(f(x)\) makes the loss function \(F(x)\) in latest round:

\[
J_{\text{asy}}(F(x) + f(x)) = E[I(y = 1)e^{-C_1 y F(x)} + I(y = -1)e^{-C_2 y F(x)}]
\]

where, \(w(x, y) = e^{-y C_i F(x)}\), \(E_w[.]\) is weighted expectation, and the expression is:

\[
E[w(x, y)] = \frac{E[w(x, y)g(x, y)]}{E[w(x, y)]}
\]

Current optimal weak classifier \(f(x)\) is chosen with Newton modified method similar to Gentle AdaBoost algorithm, and expressed as this:

\[
f(x) = C_1 P_w(y = 1|x) - C_2 P_w(y = -1|x)
\]

where, \(P_{w}(y = 1|x)\) and \(P_{w}(y = -1|x)\) are weight cumulative distributions of positive and negative training samples, respectively. The transformation formula of weight for every round is:

\[
w_i^{(m+1)} = w_i^{(m)} e^{-C_i y_i f(x)}
\]

### III. DYNAMIC OPTIMIZE STRUCTURE OF LSTM

After data preprocessing, a dynamic optimization structure of LSTM method is proposed for short-term traffic flow prediction. In this section, we introduce LSTM model, and propose an improved chaotic particle swarm optimization algorithm, which can dynamic optimize hidden layer structure of LSTM for adapting the change of traffic flow data.
A. LSTM MODEL

In this paper, LSTM is used for predicting short-term traffic flow. LSTM model is composed of input layer, hidden layer and output layer. Distinguished from traditional RNN, the hidden layer structure of LSTM increases the memory block, which can store and deliver information in a long time. Each memory block consist of input gate, output gate, forget gate and memory cell. The structure of memory block in LSTM is shown in Figure 1. Under the action of the activation function, input gate, output gate and forget gate can produce a number between 0 and 1, which are used for controlling switch of the door. Input gate control if information of input layer is passed to the hidden layers, close is the switch of the door. Input gate control if information of input layer into hidden layers, close is the switch of the door. Input gate control if information of input gate, forgetting gate, output gate and cell state respectively. \( b_\psi, b_\phi, b_o, b_c \) are bias parameters of input gate, forgetting gate, output gate and cell state respectively. \( \sigma \) is the standard logistics sigmoid function and \( \tanh \) is the hyperbolic function, they are defined as follows:

\[
\sigma(x) = \frac{1}{1 + e^{-x}}
\]

\[
\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}
\]

B. IMPROVED CHAOTIC PARTICLE SWARM OPTIMIZATION ALGORITHM

In order to promote accuracy, an improved chaotic particle swarm optimization (CPSO) algorithm is proposed to optimize hidden layer structure in LSTM model. Minimum mean absolute error of predicted value and true value is set as the optimization objective function.

In traditional particle swarm optimization, positions and masses of initial particles are generated so randomly that particles are easily trapping in local optimum and fail to get global optimum [33]. In this paper, PSO is improved. Firstly, chaotic sequences instead of random selection produce uniformly distributed initial particles, improving their quality. In addition, chaos perturbation is utilized to help particles escape from local optimum avoiding premature. CPSO implementation steps are as follows:

1) CHAOTIC INITIALIZATION

For standard PSO algorithm, some of randomly generated initial particles are far away from the global optimum, severely affecting convergence rate [34]. Chaotic sequences initial particles are close to global optimum overcoming the drawback of slow convergence. Firstly, chaotic sequences are generated with Logistic map as follows:

\[
z_{n+1} = \mu z_n (1 - z_n), \quad n = 1, 2, \ldots, N - 1
\]

where, \( \mu \) is control parameter, \( n \) is iterations, \( z_n \) is the chaos of nth iteration, \( z_1 \in [0, 1], \mu \in (2, 4] \).

Steps of chaos initialization are as follows:

**Step 1:** randomly generate a \( n \)-dimension vector \( z_1 = (z_{11}, z_{12}, \ldots, z_{1n}) \), and obtain \( z_1, z_1, \ldots, z_N \) according to equation 23.

**Step 2:** set initial positions of every particles with chaotic sequences \( z = (z_1, z_2, \ldots, z_N) \), and by mapping vectors of chaotic sequences get particles initial positions within optimum scope, as follows:

\[
x_{ij} = p_{\text{min}} + (p_{\text{max}} - p_{\text{min}})z_{ij}
\]

where, \( p_{\text{min}} \) and \( p_{\text{max}} \) are the minimum and maximum within optimum scope, respectively, \( i = 1, 2, \ldots N, j = 1, 2, \ldots n, x_i \) is initial position of particle.

**Step 3:** for every particle calculate fitness value of objective function on \( x_i \). Choose \( m \) particles with optimum fitness values and generate particle group \( x_i = (x_{1i}, x_{2i}, \ldots, x_{mi}) \).

**Step 4:** randomly generate initial velocities of \( m \) particles \( v_i = (v_{i1}, v_{i2}, \ldots, v_{im}) \), where, \( i = 1, 2, \ldots m \).
2) CHAOTIC PERTURBATION

Chaotic perturbation helps particles escape from local optimum and prevents particles from trapping in local optimum and premature. Chaos perturbation is carried out as described below:

Step 1: randomly generate a n-dimensional vector \( \eta_i = (\eta_{i1}, \eta_{i2}, \ldots, \eta_{in}) \), \( \eta_{ii} \in [0, 1] \), \( i = 1, 2, \ldots, n \), and according to equation 23, obtain chaotic sequences.

Step 2: if range of chaotic perturbation is \([-\delta, \delta]\), map chaotic sequences in step 1 to offset positions of perturbation particles, as follows:

\[
\Delta x_j = -\delta + 2\delta \tau_{ij}, \quad j = 1, 2, \ldots, n
\]  

Offset positions of chaos perturbation particles are represented as \( \Delta x = (\Delta x_1, \Delta x_2, \ldots, \Delta x_n) \).

Step 3: according to equations below, calculate particle positions of normal operation at next time and that after increasing chaos perturbation, respectively:

\[
x_{t+1}^{k} = x_{t}^{k} + v_{i}^{k+1}
\]  
\[
\Delta x_{t}^{k} = x_{t}^{k} + v_{i}^{k+1} + \Delta x
\]

Step 4: calculate fitness values of objective function when particle is at position of \( x_{t}^{k+1} \) and \( \Delta x_{t}^{k+1} \). If fitness value of objective function at \( \Delta x_{t}^{k+1} \) is better than that at \( x_{t}^{k+1} \), then replace \( x_{t}^{k+1} \) with \( \Delta x_{t}^{k+1} \).

By chaotic initialization, perturbation and local search, modified CPSO algorithm improves qualities of initial particles and enhances the capacity of escaping from local optimum and searching for global optimum.

IV. DYNAMIC OPTIMIZATION AGACS-LSTM SHORT-TERM TRAFFIC FLOW PREDICTION

Collected traffic flow data is classified with Asym-Gentle Adaboost with Cost-sensitive SVM. Outliers are removed, while real data is reserved and used to train LSTM model. And an improved CPSO is used for dynamic optimizing hidden layer structure of LSTM model. Therefore, we build a short-term traffic flow prediction model combining AGACS with CPSO-LSTM, which is named dynamic optimization AGACS-LSTM (AGACS-CPSO-LSTM) short-term traffic flow prediction model.

Implement steps of AGACS-CPSO-LSTM short-term traffic flow prediction model are as follows:

Step 1 (Initialization):

1. Input training sample set \( E = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), \( x_i \in X, y_i \in [-1, 1], X \) is sample space, \( Y = 1 \) means it is positive sample, or real data. \( Y = -1 \) is negative sample, or outlier. Set kernel function parameters \( \sigma \) and \( \nu \) for CS-SVM weak classifier.

2. Initialize sample weight \( w_i = 1/2m, 1/2n \), where, \( m \) and \( n \) are sizes of positive and negative sample, respectively.

3. Set cost sensitivity coefficient of Asym-Gentle Adaboost, \( C_1, C_2, C_1 > C_2, C_1, C_2 \in (0, 1) \).

Step 2 (Training Classifier):

1. Choose cost weight coefficient \( \gamma \) of CS-SVM weak classifier and train the classifier.

2. Calculate classification error rate of unbalanced sample \( h_t \): \( \varepsilon_{\text{asym}}(C_1, C_2) = C_1 N_{\text{FN}} + C_2 N_{\text{FP}} \), where, \( N_{\text{FN}} \) is number of misclassified positive samples, \( N_{\text{NP}} \) is number of misclassified negative samples.

3. For every CS-SVM weak classifier:
   a. Divide sample space into several subspaces \( X_1, \ldots, X_n \).
   b. Calculate the formula below for subset of training sample:

\[
w_i^l = P(x_i \in X_j, y_i = l) = \sum_{i=x_i \in X_j \land y_i=l} \text{E}(i), \quad l = \pm 1
\]

c. Output optimal weak classifier in every sample subspace \( X_j \):

\[
h_j = \frac{C_1 P_w(y = 1|x) - C_2 P_w(y = -1|x)}{C_1^2 P_w(y = 1|x) - C_2^2 P_w(y = -1|x)}
\]

d. Calculate normalization factor:

\[
Z_j = \sum_i D_t(i) \exp(-C_j y f_j(x_i))
\]

4. Pick the minimum \( Z_i \). According to \( h_i^l(x) = \arg\min Z_i \), \( h_i^l(x) \) is the best weak classifier in this iteration.

5. Adjust and normalize sample distribution:

\[
D_{t+1}(i) = \frac{D_t(i) \exp(-C_j y h_t(x_i))}{Z_i}
\]

Step 3: export the final strong classifier: \( H(x) = \text{sign} (\sum_h(x)) \)

Step 4: set structure and hyper-parameter of LSTM model, including node number in every layer, excitation function, learning rate, loss function, optimization function, batch number, iteration number and so on.

Step 5: train LSTM model with data preprocessed in AGACS. With modified chaos particle swarm optimization, dynamically optimize parameters of hidden layer based on objective function and update layer connection weights according to error signal. Stop training until permissible error or maximum number of iterations is achieved.

Step 6: obtain predictions for short time traffic flow by trained LSTM model with real traffic flow data preprocessed in AGACS.

V. EXPERIMENTS AND RESULTS ANALYSIS

A. SAMPLE COLLECTION AND ANALYSIS

This experiment select 15 days traffic flow data from Dec 16, 2019 to Dec 30, 2019, which come from a road sense coil in Fuzhou city china. Sampling interval is 5 min and the total 288 * 15 sets of data are collected. According to the literature [35], traffic flow characteristics are different among weekday daytime, weekday nighttime, weekend daytime and weekend nighttime. Therefore, all data is divided into four categories according to the four time periods. In these sets, daytime is from 6 a.m. to 8 p.m and nighttime is from 8 p.m.
to 6 a.m. next day; weekday is Monday to Friday and weekend is Saturday to Sunday. Traffic flow data from Dec 16, 2019 to Dec 28, 2019 is training sets, which are used to train structure of model, and the data from Dec 29, 2019 (weekend) to Dec 30, 2019 (weekday) is testing sets for verifying the performance of the prediction algorithm.

B. EVALUATE AGACS DATA PREPROCESSING PERFORMANCE

In this experiment, detection accuracy rate, error detection rate and false alarm rate are used to evaluate the data preprocessing performance of AGACS algorithm [36]–[38]. The formulas are as follows:

\[
D_{\text{acc}} = \frac{T_{CN} + T_{CG}}{T_{CN} + T_{CG} + T_{EN} + T_{EG}}
\]

(28)

\[
R_{FP} = \frac{T_{EN}}{T_{EN} + T_{CG}}
\]

(29)

\[
R_{FN} = \frac{T_{EG}}{T_{EG} + T_{CN}}
\]

(30)

where, \(D_{\text{acc}}\), \(R_{FP}\) and \(R_{FN}\) are detection accuracy rate, error detection rate and false alarm rate respectively. \(T_{CN}\) and \(T_{EN}\) denote outlier data is detected and undetected respectively, \(T_{CG}\) and \(T_{EG}\) denote normal data is detected and undetected respectively.

Test samples are 4320 sets of traffic data from Dec 16, 2019 to Dec 30, 2019, AGACS algorithm and the other three classic methods (CS-SVM, Asym-Gentle Adaboost and SVM) are used to detect the outlier data in the test samples respectively, and compare the performance of the four models. In three classic algorithms, CS-SVM and Asym-Gentle Adaboost (AG-Adaboost) belong to the cost-sensitive classification method, and that SVM is general classification method. The parameters \(C_1\) and \(C_2\) in AGACS are 0.7 and 0.3 respectively, and the misclassification weighted value \(\gamma\) in AGACS, CS-SVM and AG-Adaboost separately select 0.2, 0.4, 0.6, 0.8, 1 five kinds of number, and comparing the performance of these methods in different parameters. Figures 2 - 4 denote in the five kinds of misclassification weighted value, three kinds of cost-sensitive classification methods get the detection accuracy rate, error detection rate and false alarm rate. Table 1 shows the optimal evaluation result of the four different algorithms respectively.

| Algorithm            | Optimal \(R_{FP}\) | Optimal \(R_{FN}\) | Optimal \(D_{\text{acc}}\) |
|----------------------|--------------------|--------------------|--------------------------|
| AGACS                | 3.6                | 5.6                | 93                       |
| CS-SVM               | 8.8                | 9.3                | 81                       |
| AG-Adaboost          | 5.3                | 7.7                | 86                       |
| SVM                  | 16.7               | 18.1               | 72                       |

From the result of Table 1, in the same data sets, the performance of different pretreatment methods is significant differences, and SVM is the most inefficient in these four algorithms. Because SVM is the only one which not used cost-sensitive to dispose the outlier data, it cannot adapt the condition of significant unbalanced between the number of outlier data and normal data. Therefore, we make use of the cost-sensitive to handle the outlier data is efficient. Figure 2, Figure 3 and Figure 4 show the misclassification weighted value is a very important parameter in cost-sensitive classification methods.
classification algorithms, and it is greater impact the performance of methods. Why it is so important in cost-sensitive classification algorithms? Because it can controls the cost sensitivity of classifier and the punishment value of misclassification, so that the algorithm pays more attention to negative samples which are few numbers, and classifies them more accurately. In this experiment, the optimal misclassification weighted value is 0.8. Compare the performance of these three cost-sensitive classification methods, the proposed AGACS algorithm is the best of all, and AG-Adaboost is better than CS-SVM. In the optimal detection accuracy rate, AGACS is higher than AG-Adaboost 7%, and in the optimal error detection rate and false alarm rate, AGACS is less than AG-Adaboost 1.7% and 2.1% respectively. This shows that AGACS can better focus on and eliminate the outlier data. The proposed algorithm improves weak classifier and combination strong classifier simultaneously, which make them cost-sensitive, hence it gets the best result under the unbalanced data classification.

C. EVALUATE THE EFFECTIVE OF DYNAMIC OPTIMIZATION MODEL STRUCTURE

In this paper, we propose a method to dynamic optimize the structure of AGACS-LSTM model based on chaotic particle swarms optimization (CPSO). In order to verify the effect of this method, this experiment set up the parameters of CPSO, and employ the prediction accuracy rates to compare the propose method with non-optimized and other two state of the art algorithms (particle swarms optimization (PSO), genetic algorithm (GA)). The results are shown in Figure 5.

Figure 5 shows the predict results of weekday daytime, weekday nighttime, weekend daytime and weekend nighttime four different time periods. In this diagram, we can see that if non-optimized (Iteration is 0), the highest accuracy rates in weekday daytime, weekday nighttime, weekend daytime and weekend nighttime are 0.85, 0.74, 0.83 and 0.71, respectively. By optimizing, predict results of all time periods are improved. This result shows that dynamic optimize the structure of AGACS-LSTM model can improve the predict performance. After the dynamic optimize of CPSO, the highest accuracy rates in above four different time periods have been raised to 0.94, 0.83, 0.92 and 0.81. This result also suggest that CPSO outperforms the other two state of the art algorithms to improve the predict accuracy rates. And in CPSO, the highest accuracy rate appears in about 20 iterations, but in PSO and GA, the highest accuracy rates appear in about 50 and 80 iterations respectively. This result can illustrate the convergence of CPSO is superior to the other two algorithms. It’s worth mentioning that CPSO make the accuracy rates obvious improvement constantly until the highest accuracy rate appeared. But in the other two algorithms, the accuracy rates often fluctuate. This phenomenon explains the global search ability of proposed CPSO outperforms the other two
state of the art algorithms, it can jump from the local optimal to the global optimal successfully.

**D. EVALUATES THE PERFORMANCE OF THE DYNAMIC OPTIMIZATION AGACS-LSTM MODEL**

This section evaluates the performance of the proposed prediction method which named dynamic optimization AGACS-LSTM short-term traffic flow prediction (AGACS-CPSO-LSTM). Firstly, set the parameters of the algorithm and select indicators for performance evaluation, we compare the performances of AGACS-CPSO-LSTM, dynamic optimization LSTM (CPSO-LSTM) and AGACS-LSTM model, which is used for evaluating the effectiveness of data preprocessing and optimization model structure to short-term traffic flow prediction. Furthermore, we compare our AGACS-CPSO-LSTM model with other three state of the art traffic flow prediction algorithms, which demonstrate the advantages of the proposed method.

The initial parameters of AGACS-CPSO-LSTM are given as follows: the cost asymmetry weight coefficient $\gamma$ in CS-SVM weak classifier is 0.8, the cost asymmetry weight in strong classifiers $C_1$ and $C_2$ are 0.7 and 0.3 respectively. The initial number of input nodes in the LSTM model is 60, and the number of layers of the hidden layer structure is selected dynamically by the optimization algorithm according to the value of the target function.

In order to evaluate the short-term traffic flow prediction performance of the algorithms, this experiment choose MRE, MAE and RMSE three indicators [39], [40], and the formula are as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$$  \hspace{1cm} (31)

$$MRE = \frac{1}{n} \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{y_i}$$  \hspace{1cm} (32)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$$  \hspace{1cm} (33)

where, $y_i$ is the practical traffic flow, $\hat{y}_i$ represent the predicted traffic flow, $n$ is the number of samples.

Four categories collect data sets in section V.A are used for training models, and according to the previous one hour traffic flow predict the next 15 minutes traffic flow. The proposed AGACS-CPSO-LSTM, CPSO-LSTM and AGACS-LSTM...
predict traffic flow data of weekday daytime, weekday nighttime in Dec 30, 2019 (weekday) and weekend daytime, weekend nighttime in Dec 29, 2019 (weekend), respectively. Then, we compare the true data in testing sets and the prediction data. The results are shown in Figure 6 and the evaluation indexes are shown in Table 2.

From Figure 6, we can see there are different traffic flow characters in weekday daytime, weekday nighttime, weekend daytime and weekend nighttime. In these four different time periods, the CPSO-LSTM is the least effectively in the three short-term traffic flow prediction algorithms. That is because different from AGACS-CPSO-LSTM and AGACS-LSTM, collecting data is used for training CPSO-LSTM model directly, without data preprocessing. Some predictions are badly wrong such as 10:00, 14:00 and 17:00 in weekday daytime, and these results may influence by outlier data...
TABLE 2. Performance of the three models at weekday time.

| model          | weekday daytime | weekday nighttime |
|----------------|-----------------|-------------------|
|                | MRE  | MAE  | RMSE  | MRE  | MAE  | RMSE  |
| AGAC-S-CPSO-LSTM | 0.058 | 29.02 | 41.33 | 0.151 | 8.97  | 14.53 |
| AGAC-S-LSTM     | 0.078 | 34.81 | 48.97 | 0.189 | 10.43 | 16.21 |
| CPSO-LSTM       | 0.095 | 39.66 | 57.12 | 0.212 | 11.67 | 18.46 |

TABLE 3. Performance of the three models at weekend time.

| model          | weekend daytime | weekend nighttime |
|----------------|-----------------|-------------------|
|                | MRE  | MAE  | RMSE  | MRE  | MAE  | RMSE  |
| AGAC-S-CPSO-LSTM | 0.072 | 24.43 | 33.24 | 0.163 | 9.53  | 15.32 |
| AGAC-S-LSTM     | 0.091 | 28.24 | 39.68 | 0.192 | 10.84 | 20.08 |
| CPSO-LSTM       | 0.117 | 32.55 | 46.22 | 0.224 | 13.26 | 22.41 |

which exist in the collecting data. Therefore, we preprocess collecting data to get rid of outlier data can improve the performance of forecasting. AGAC-S-LSTM can eliminate outlier data, and the predict results are better than CPSO-LSTM. However, it is still inferior to AGAC-CPSO-LSTM. The reason is that structure of AGACS-LSTM is too fixed to adapt the change of traffic flow characteristics at different times.

Table 2 and Table 3 indicate the proposed method AGAC-CPSO-LSTM is most accurate in these three models at all times. The results illustrate that preprocess collecting data and dynamic optimize model structure are conducive to improve the performance of algorithm, and the propose methods of eliminate outlier data and optimize model structure in this paper are effective and feasible.

In the same training and testing sets, we compare our proposed model with the three classical algorithms, which are deep belief networks (DBN), convolutional neural networks (CNN) and recurrent neural networks (RNN) traffic flow prediction algorithms. Firstly, we eliminate outlier data in four categories collect data sets in Section V.A by AGACS model, and then, train models and according to the previous one hour traffic flow predict the next 15 minutes traffic flow. The proposed AGAC-CPSO-LSTM and the above three classical algorithms predict traffic flow data of weekday daytime weekday nighttime in Dec 30, 2019 (weekday) and weekend daytime weekend nighttime in Dec 29, 2019 (weekend), respectively. Then, we compare the true data in testing sets and the prediction data. The results are shown in Figure 7.

As shown in Figure 7, we can see that the performance of our model outperforms the other three classical deep learning traffic flow prediction algorithms. In these four different time periods, AGAC-CPSO-LSTM can obtain the lowest MRE, MAE and RMSE in 5, 10 and 15min short-term traffic flow prediction. Therefore, the proposed method is robust for different short-term prediction time intervals. And then, AGAC-CPSO-LSTM is more accurate in daytime than that in nighttime, it indicates that this method is more effective in high traffic flow condition. Compare the results of 5, 10 and 15min short-term traffic flow prediction, we can see that the shorter the prediction time, the better the prediction effect, which shows that the shorter the time interval, the closer the temporal continuity between forecast data and historical data. AGAC-CPSO-LSTM can capture the time continuity of traffic flow well, so as to achieve high-precision short-term traffic flow prediction.

VI. CONCLUSION AND FUTURE WORK

This paper proposed a dynamic optimization LSTM model based on data preprocessing for predicting short-term traffic flow. Consider the influence of the outlier data, AGAC classification algorithm is proposed to divide the collected data into outlier data and normal data. And then, eliminate outlier data and train LSTM by normal data. In order to promote the generalization performance of LSTM, an improve CPSO is used for dynamic optimizing hidden layer structure of LSTM model. The efficiency of the proposed method is proved with real traffic flow data, the experimental results show that the prediction accuracy of AGAC-CPSO-LSTM is better than the other three state-of-the-art algorithms, and the practicability of the model is improved significantly. In future, there are still many potentials work to do. In this paper, we only consider history traffic flow data in prediction node, and did not consider the influence of upstream and downstream traffic flow data. Therefore, our next research will focus on the rule of spatial evolution about traffic flow, and consider the link between the adjacent intersections and forecast section in traffic flow prediction.

REFERENCES

[1] J. Xue and Z. Shi, “Short-time traffic flow prediction based on chaos time series theory,” J. Transp. Syst. Eng. Inf. Technol., vol. 8, no. 5, pp. 68–72, Oct. 2008.
[2] Y. Kamarianakis, H. Oliver Gao, and P. Prastacos, “Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions,” Transp. Res. C, Emerg. Technol., vol. 18, no. 5, pp. 821–840, Oct. 2010.
[3] T. Thomas, W. Weijermars, and E. V. Berkum, “Predictions of urban volumes in single time series,” IEEE Trans. Intell. Transport. Syst., vol. 11, no. 1, pp. 71–80, Mar. 2010.
[4] B. L. Smith, B. M. Williams, and R. K. Oswald, “Comparison of parametric and nonparametric models for traffic flow forecasting,” Transp. Res. C, Emerg. Technol., vol. 10, no. 4, pp. 303–321, Aug. 2002.
[5] W.-C. Hong, Y. Dong, F. Zheng, and C.-Y. Lai, “Forecasting urban traffic flow by SVR with continuous ACO,” Appl. Math. Model., vol. 35, no. 3, pp. 1282–1291, Mar. 2011.
[6] D. Wei and H. Liu, “An adaptive-margin support vector regression for short-term traffic flow forecasting,” J. Intell. Transp. Syst., vol. 17, no. 4, pp. 317–327, Oct. 2013.
[7] A. Cheng, X. Jiang, Y. Li, C. Zhang, and H. Zhu, “Multiple sources and multiple measures based traffic flow prediction using the chaos theory and support vector regression method,” Phys. A, Stat. Mech. Appl., vol. 466, pp. 432–434, Jan. 2017.
