Research Article

Short-Term Traffic Prediction considering Spatial-Temporal Characteristics of Freeway Flow

Jiaqi Wang, Yingying Ma, Xianling Yang, Teng Li, and Haoxi Wei

Department of Transportation Engineering, South China University of Technology, 381 Wushan Road, Guangzhou 510641, China

Correspondence should be addressed to Yingying Ma; mayingying@scut.edu.cn

Received 16 April 2021; Revised 15 July 2021; Accepted 7 September 2021; Published 13 October 2021

Academic Editor: Feng-Jang Hwang

Copyright © 2021 Jiaqi Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a short-term traffic prediction method, which takes the historical data of upstream points and prediction point itself and their spatial-temporal characteristics into consideration. First, the Gaussian mixture model (GMM) based on Kullback–Leibler divergence and Grey relation analysis coefficient calculated by the data in the corresponding period is proposed. It can select upstream points that have a great impact on prediction point to reduce computation and increase accuracy in the next prediction work. Second, the hybrid model constructed by long short-term memory and K-nearest neighbor (LSTM-KNN) algorithm using transformed grey wolf optimization is discussed. Parallel computing is used in this part to reduce complexity. Third, some meaningful experiments are carried out using real data with different upstream points, time steps, and prediction model structures. The results show that GMM can improve the accuracy of the multifactor models, such as the support vector machines, the KNN, and the multi-LSTM. Compared with other conventional models, the TGWO-LSTM-KNN prediction model has better accuracy and stability. Since the proposed method is able to export the prediction dataset of upstream and prediction points simultaneously, it can be applied to collaborative management and also has good potential prospects for application in freeway networks.

1. Introduction

Intelligent transportation system (ITS) has become an effective way to reduce pollution and improves the performance of freeways, while the short-term traffic flow prediction is an important part to support the smart management and control of freeways. The trend of short-term traffic flow prediction is changing from parametric statistical models to nonparametric models and mixed models. Time-series methods were widely used in parametric statistical models, including exponential smoothing [1–3], moving average [4, 5], and autoregressive integrated moving average (ARIMA) model [6–8]. Kalman filtering was also used for traffic flow prediction, such as adaptive Kalman filter [9–11], hybrid dual Kalman filter [12], and noise-identified Kalman filter [13]. With the rapid development of ITS and improvement of data quality, more nonparametric prediction methods are used in the prediction of traffic flow. K-Nearest Neighbor (KNN) nonparametric regression, a nonlinear prediction method, was used to calculate Euclidean distance to find the nearest neighbor for prediction [14]. The improved Bayesian combination model was proposed to increase the accuracy of prediction [15]. Support vector machines (SVM) were also used considering the weak sensitivity to outliers [16]. The combined algorithm based on wavelet packet analysis and least square support vector machines was used to resolve the uncertainty and randomness of data [17]. Particle swarm optimization (PSO) and other optimization algorithms were applied to SVM because of small model calculation and good prediction performance [18]. With the development of artificial intelligence (AI), deep learning models have been widely used in traffic prediction. Smith and Demetsky [19] used back-propagation (BP) neural network to do the prediction. Optimization algorithms such as PSO and genetic algorithm (GA) were also applied to BP, and the effect is obvious [20, 21]. Recurrent neural network (RNN) can realize long-term memory calculation and was used in prediction, but it
had the problem of gradient explosion [22]. Long short-term memory (LSTM) network was proposed to solve it by using a forget gate [23, 24], which was not only used in natural language processing [25], for example, language generation [26], text classification [27], and phoneme classification [28], but also in prediction fields, such as short-term traffic flow prediction [29], housing load prediction [30], and pedestrian trajectory prediction [31]. Furthermore, improvements and combinations with other models have been proposed in many fields, from application in large-scale data problems [32] to the prediction of traffic flow, such as using GA to optimize the LSTM hyperparameters to get better performance [33]. The comparison of typical machine learning models is shown in Table 1.

Deep learning models are widely used in traffic flow prediction, especially in short-term prediction [41]. However, traffic flow has strong spatial-temporal characteristics on time series [42, 43]. More attention was paid to this characteristic in recent years’ research of short-term traffic flow prediction [44–46]. Luo et al. [40] proposed a spatial-temporal traffic flow prediction model with KNN and LSTM to screen highly correlated upstream points and produced the prediction. Ma et al. put forward a method to select input data for daily traffic flow forecasting through contextual mining and intraday pattern recognition [47] and produced the daily traffic flow forecasting with CNN and LSTM [48]. Supervisory learning was used to mine the relationship between the factors of historical data and current traffic flow to train the predictor in advance so as to reduce the predicting time [49]. In addition, the match-then-predict method [50] and the fuzzy hybrid framework [51] with dynamic weights by mining spatial-temporal correlations were both proposed. Attention mechanisms were also combined in LSTM to increase the accuracy of prediction [52]. These methods that combine various factors using attention mechanism can reasonably allocate limited resources, increase the efficiency, and reduce computation.

In this paper, we propose the short-term traffic flow prediction model considering the spatial-temporal characteristics using LSTM and KNN under the concept of attention mechanism. First, the Gaussian mixture model (GMM) is used to select the upstream detection points to produce the prediction. Two parameters are used for the classification: one is the Kullback–Leibler Divergence (KL), also known as the relative entropy, which reflects the difference in the distribution of two datasets through approximate calculations, especially for large-sample traffic data. The other is the grey relation analysis (GRA) coefficient, which reflects the correlation between two groups of normalized data after similarity analysis. Second, the hybrid model of LSTM and KNN is proposed to produce the prediction using the selected data. LSTM is used to predict the traffic flow of upstream points as the training dataset of KNN. To solve the problem of time lag, input time of upstream data is changed in the model according to the space distance between the input point and the prediction point and the average speed of traffic flow. Moreover, transformed grey wolf optimizer (TGWO) is used to optimize key parameters, and Savitzky-Golay (SG) filter smoothing is used to reduce the noise in the model to improve the performance. The proposed TGWO-LSTM-KNN prediction model in this paper gives greater consideration to the spatial-temporal characteristic of freeway traffic flow to improve the accuracy of prediction and reduces the complexity of computation by selecting and preprocessing input data.

The rest of this paper is organized as follows. Section 2 introduces the methodology of the proposed model. Section 3 carries out the experiments and analysis of the proposed model with real-world traffic flow data. Section 4 presents the conclusions and the prospects of the research. The abbreviations used in the rest of the paper are listed in Table 2.

2. Methodology

2.1. Framework. This paper proposes TGWO-LSTM-KNN with the GMM classification model, which includes two parts: data preparation and prediction. GMM is used to choose the input data in the data preparation part considering the spatial-temporal characteristics of freeway traffic flow, while the prediction part is composed of LSTM parallel computing module, KNN module, and TGWO module. The framework of the proposed model is shown in Figure 1.

2.1.1. Data Preparation. In the freeway network, the traffic flow of upstream correlates with the prediction point flow, which is considered as spatial correlation. Moreover, the traffic flow of the prediction point changes over time inter- and intraday, and each specific period may have different patterns, such as the morning peak hour and the off-peak hour. Therefore, time series are divided into different parts according to the flow patterns, which will help to improve the accuracy of prediction. The temporal characteristic of prediction point flow is observed through a flow chart. Then set the midpoint of two adjacent extreme points and complete the time series division task.

Related upstream sections are analyzed and selected using GMM binary classification. In this paper, two parameters are used as the classification criteria. One is the KL divergence, which is a commonly used method in information science to quantify the difference between two datasets. In a large-sample traffic dataset with complex distribution, the difference can be simply and quickly reflected. The other is the GRA coefficient, which can analyze the linear similarity between two datasets through a small amount of data. These two parameters can well reflect the correlation of upstream sections and predicted sections. The steps of the classification part are as follows:

Step 1: use time step and speed to determine the space range. Note the upstream points within the scope as $O_{1:m}$. 
Step 2: divide day time into $T_1 \sim T_{q'}$. 
Step 3: construct dataset. Divide the dataset into working days and nonworking days. Change the input time of upstream data to meet the time lag of the prediction point considering the distance and travel speed.
Step 4: calculate the KL divergence and the GRA coefficient in $T_i$ of working days and nonworking days.

Step 5: input the KL divergence and the GRA coefficient into GMM for binary classification.

$O_1 \sim O_m$ are divided into two groups in each $T_i$. The group of points with the KL divergence close to 0 and the GRA coefficient close to 1 is used as the strongly related section of the prediction point for the next prediction.

2.2. Data Preparation. There are three steps in data preparation: determination of spatial scope, time-series division, and GMM classification.

2.2.1. Determination of Spatial Scope. Since the time step of short-term traffic flow prediction ($T_{step}$) is usually less than one hour and the highway speed ($V$) is limited, the radius range of spatial scope can be calculated:

$$R = T_{step} \times V. \quad (1)$$

The accesses and ramps within the radius of the prediction point centered are selected as upstream points.

2.2.2. Time Series Division. There are random fluctuations in daytime traffic flow, as shown in Figure 2. SG is a method to smooth data based on local least-squares polynomial approximation proposed by Savitzky and Golay [53], which is used to get the extreme traffic value points of the prediction point ($D_0^p$) (see Figure 2). Set the point in the middle of the

| Table 1: Comparison of machine learning models in short-term traffic flow prediction. |
|-----------------------------------|---------------------------------|---------------------------------|
| **The basic model**              | **Model performance**           | **Improved models**             |
| Backpropagation neural network (BP) | Advantages [34]                | GA-BP [21], PSO-BP [20],       |
|                                   | (1) Self-adaptive self-learning | EEMD-IAGA-BP [35]              |
|                                   | (2) Strong nonlinear mapping ability |                                |
|                                   | (3) Strong generalization ability |                                |
|                                   | Disadvantages [34]              |                                |
|                                   | (1) Slow convergence speed down  |                                |
|                                   | (2) Easy to fall into the local optimal solution | |
| Least-squares support vector machines (LSSVM) | Advantages [36] | W-LSSVM [17], PSO-SVM [18],    |
|                                   | (1) Suitable for small datasets  | EMD-GPSO-SVM [37]              |
|                                   | (2) Simple and convenient        |                                |
|                                   | (3) Strong nonlinear generalization ability |                                |
|                                   | Disadvantages [36]              |                                |
|                                   | (1) Unable to handle large datasets |                                |
|                                   | (2) Sensitive to missing values  |                                |
| Long short-term memory (LSTM)     | Advantages [38]                 | GA-LSTM [33], SVD-PSO-LSTM [39], |
|                                   | (1) High precision               | KNN-LSTM [40]                  |
|                                   | (2) Solve the problem of RNN gradient |                                |
|                                   | Disadvantages [38]              |                                |
|                                   | (1) Time-consuming calculation   |                                |
|                                   | (2) Large consumption of hardware resources |                                |
|                                   | (3) Too many parameters, easy to overfit |                        |

| Table 2: Abbreviation table for algorithm required in this paper. |
|---------------------------------------------------------------|
| **Name**          | **Abbreviation** |
| Long short-term memory                                      | LSTM |
| K-nearest neighbor algorithm                                 | KNN |
| Kullback-Leibler                                             | KL |
| Grey relation analysis                                       | GRA |
| Gaussian mixture model                                       | GMM |
| Expectation maximization                                     | EM |
| Transformed grey wolf optimization                           | TGWO |
| Support vector machines                                      | SVM |
| Backpropagation neural network                               | BP |
| Root mean square error                                       | RMSE |
| Mean absolute error                                          | MAE |
| Mean absolute percentage error                               | MAPE |

reduce the time consumption of prediction. Also, to improve the performance of LSTM-KNN, TGWO is used to optimize the parameters of LSTM and KNN.

Step 1: use TGWO to optimize the steps and epochs in LSTM, K value in KNN.

Step 2: multithread LSTM parallel computing is used to reduce calculation by ignoring the relationship among upstream points. Each $O_i$ is input into the corresponding LSTM module and then the output $P_i$ set and $D_0$ together form a new dataset.

Step 3: input the dataset into the KNN module to predict the traffic flow and output $D_0^p$.

2.1.2. Prediction. The prediction part consists of three modules, which are LSTM module, KNN module, and TGWO module. KNN is selected at the bottom of the model considering the spatial features of freeway flow with the advantages of fast calculation speed and no lag. LSTM is used to predict the short-term traffic flow of upstream sections and then put the prediction results of upstream sections into KNN to predict the prediction point traffic flow. Because the relationship among upstream points is ignored in the model, multithread LSTM parallel computing (LSTMs) is used to reduce the time consumption of prediction. Also, to improve the performance of LSTM-KNN, TGWO is used to optimize the parameters of LSTM and KNN.

Step 4: calculate the KL divergence and the GRA coefficient in $T_i$ of working days and nonworking days.

Step 5: input the KL divergence and the GRA coefficient into GMM for binary classification.

$O_1 \sim O_m$ are divided into two groups in each $T_i$. The group of points with the KL divergence close to 0 and the GRA coefficient close to 1 is used as the strongly related section of the prediction point for the next prediction.
two adjacent extreme values, and each time series between two middle points is a time part \( (T_i) \) (see Figure 3).

2.2.3. SG Calculation Method. SG includes two parameters, which are the window length \( n \) and the order number \( k \). For the window length \( n \), with the increasing of \( n \), the deviation between the processed data and the real data increases, and also the smoothness. For order number \( k \), with the increasing of \( k \), the deviation between the processed data and the real data decreases, and also the smoothness. According to the characteristics of highway traffic flow and existing research, the choice of \( n \) and \( k \) in this paper is 31 and 1, respectively.

Input upstream points data \( O_m = (o_m(1), o_m(2), \ldots, o_m(\text{x})) \), where \( \text{x} \) denotes the length of data. Select the window length \( n \) and order number \( k \). The data in a window

\[
O_m = (o_m(1), o_m(2), \ldots, o_m(n)).
\]

The fitting polynomial is obtained using the \( k-1 \) least square method as follows:

\[
o_m^{k-1}(1) = a_0 + a_1 o_m(1) + a_2 o_m(1)^2 + \ldots + a_{k-1} o_m(1)^{k-1}.
\]

Then form \( n \) equations to form \( k \) element equations. If \( n > k \), equation has a solution, then

\[
\begin{pmatrix}
o_m^{k-1}(1) \\
o_m^{k-1}(2) \\
\vdots \\
o_m^{k-1}(n)
\end{pmatrix} =
\begin{pmatrix}
1 & o_m(1) & \cdots & o_m(1)^{k-1} \\
1 & o_m(2) & \cdots & o_m(2)^{k-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & o_m(n) & \cdots & o_m(n)^{k-1}
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
\vdots \\
a_{k-1}
\end{pmatrix}
+ \begin{pmatrix}
e_0 \\
e_1 \\
\vdots \\
e_{k-1}
\end{pmatrix},
\]

\[
O_m = o_m^{k-1} \cdot A_{k \times 1} + E_{n \times 1}.
\]

The matrix is expressed as

\[
A = \text{the least square fitting solution of different windows, and the value } O_m \text{ is as follows:}
\]

\[
O_m = O_m^t \cdot A.
\]

Smoothing dataset is \( O_m' = (o_m'(1), o_m'(2), \ldots, o_m'(n)) \)

2.2.4. GMM Classification. Gaussian mixture model (GMM) is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters [54],
which is used to judge the correlation between upstream and predicted traffic flow, while EM is used to obtain the maximum likelihood estimation of GMM [55]. In this paper, KL-divergence and GRA coefficient are two parameters to do the binary classification.

(1) KL Divergence. KL divergence [56], which is also known as relative entropy, is broadly used as the measurement of the dissimilarity between two probabilistic models [57]. Define KL divergence between \( O_i = (o_i(1), o_i(2), \ldots, o_i(n)) \) and \( D_0 = (d_0(1), d_0(2), \ldots, d_0(n)) \) as

\[
\text{KL}(O_i, D_0) = \sum_{j=1}^{n} o_i(j) \log \frac{o_i(j)}{d_0(j)}.
\]

(5)

The closer KL divergence is to 0, the more similar the two distributions are.

(2) GRA Coefficient. GRA is a method to judge the similarity of different datasets. Compared with traditional Pearson correlation [58], it can use a smaller amount of data to reflect the linear similarity between traffic flows [59]. The steps of GRA are as follows.

Since there is little difference in the magnitude of traffic flow at the same point, the data are initialized by dividing the initial value of the flow \( a_1(1) \) and \( d_1(1) \).

\[
o_i(k)_{\text{GRA}} = \frac{o_i(k)}{a_1(1)},
\]

\[
d_0(k)_{\text{GRA}} = \frac{d_0(k)}{a_1(1)},
\]

(6)

where \( o_i(k) \) denotes traffic flow of upstream point \( O_i \) in the \( k \) time and \( d_0(k) \) denotes traffic flow of prediction point \( D_0 \) in the \( k \) time.

Define the prediction sequence \( (d_0(1)_{\text{GRA}}, d_0(2)_{\text{GRA}}, \ldots, d_0(n)_{\text{GRA}}) \) as \( D_{\text{GRA}} \) and the factor sequence \( (o_i(1)_{\text{GRA}}, o_i(2)_{\text{GRA}}, \ldots, o_i(n)_{\text{GRA}}) \) as \( O_{i_{\text{GRA}}} \).

Then calculate the relation coefficient in the \( k \) time

\[
\gamma(d_i(k)_{\text{GRA}}, o_i(k)_{\text{GRA}}) = \frac{\min_{i, k} \min_{d_i(k)_{\text{GRA}}, o_i(k)_{\text{GRA}}} |d_i(k)_{\text{GRA}} - o_i(k)_{\text{GRA}}| + \xi \max_{i, k} |d_i(k)_{\text{GRA}} - o_i(k)_{\text{GRA}}|}{\max_{i, k} |d_i(k)_{\text{GRA}} - o_i(k)_{\text{GRA}}|},
\]

(7)

where \( \xi \) denotes the coefficient to control the degree of differentiation, which is generally 0.5 [58].

Define the mean value as GRA coefficient of \( D_0 \) and \( O_i \):

\[
\text{GRA}(D_0, O_i) = \frac{1}{n} \sum_{k=1}^{n} \gamma(d_i(k), o_i(k)).
\]

(8)

\[
p(x) = \sum_{k=1}^{K=2} \frac{\theta_k}{2\pi^{n/2}|\Sigma_k|^1/2} \exp \left\{ -\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) \right\} = \sum_{k=1}^{K=2} \theta_k \mathcal{N}(x; \mu_k, \Sigma_k),
\]

(9)

where \( \mu_k \) denotes expectation, \( \Sigma_k \) denotes covariance, \( n \) denotes data dimension, \( \mathcal{N}(x; \mu_k, \Sigma_k) \) denotes \( k \) component in a hybrid model, \( \theta_k \) denotes mixture coefficient and

\[
\sum_{k=1}^{k=2} \theta_k = 1, \quad 0 < \theta_k < 1.
\]

(10)

Then use EM algorithm to calculate \( \theta_1, \mu_1, \Sigma_1, \theta_2, \mu_2, \Sigma_2 \).

EM-GMM pseudocode is as follows: (Algorithm 1)
LSTM are the parameters with great influence [25], and the coefficient K in the KNN module is also one of the key parameters. Then use the TGWO module to optimize these parameters.

### 2.3.1. LSTMs Module.

LSTM is a special RNN [24] with a forgetting gate. The sigmoid function is used to prevent gradient explosion and disappearance. The traffic flow data of upstream points are trained in different LSTM threads in parallel, which is defined as LSTMs. The data input time is also changed to reduce the lag of the module, which makes LSTM more accurate. The memory unit of the module is shown in Figure 5.

The calculation processes of the memory unit $o_m(n)$ and parameters (see Table 3) are as follows:

```plaintext
function EM-GMM(k ← 2, θ₁, μ₁, Σ₁, θ₂, μ₂, Σ₂, i ← 0)
do i ← i + 1:
calculate p(x);
E-step: $y(θ_β) ← θ_β \mathcal{N}(x_0\big|μ_n, Σ_n)/\sum_{j=1}^{K} θ_j \mathcal{N}(x_0\big|μ_j, Σ_j);$ 
M-step: $μ_κ^{new} ← 1/N \sum_{n=1}^{N} y(θ_β)x_0;$ 
$Σ_κ^{new} ← 1/N \sum_{n=1}^{N} y(θ_β)(x_0 - μ_κ^{new})^T;(x_0 - μ_κ^{new});$
$N_κ ← N/N;
$N_n ← \sum_{n=1}^{N} y(θ_κ);$ 
calculate $ln p^+ (x|θ, μ, Σ) ← \sum_{n=1}^{N} ln \{|\sum_{k=1}^{K} θ_k \mathcal{N}(x_k\big|θ_k, Σ_k)|/ ln p^+ (x|θ, μ, Σ) − ln p^+ (x|θ, μ, Σ) < T;$
return $θ_1, μ_1, Σ_1, θ_2, μ_2, Σ_2;$
end function
```

**Algorithm 1:** EM-GMM pseudocode.

![Figure 4: TGWO-LSTM-KNN with GMM classification flowchart.](image)
2.3.2. KNN Module. In this paper, the KNN model is used to predict \( D_0^p \) by using data of \( O_1 \sim O_m, P_1 \sim P_m, \) and \( D_0 \). This method not only has high efficiency and less complexity but also can meet the needs of multifactor. Euclidean distance is calculated as follows [14]:

\[
\bar{c}_n = \tanh(w_{m}o_m(n) + w_{ch}h_{n-1} + b_f), \\
i_n = \sigma(w_{io}o_m(n) + w_{ih}h_{n-1} + w_{ic}c_{n-1} + b_i), \\
f_n = \sigma(w_{fo}o_m(n) + w_{fh}h_{n-1} + w_{fc}c_{n-1} + b_f), \\
p_n = \sigma(w_{po}o_m(n) + w_{ph}h_{n-1} + w_{pc}c_{n} + b_o), \\
c_n = c_{n-1} \cdot f_n + i_n \cdot \bar{c}_n, \\
h_n = p_n \cdot \tanh(c_n).
\]

2.3.3. TGWO Module. Grey wolves algorithm was put forward by scholars Mir Jalili Australia in 2014, the Grey wolf groups according to social relations are divided into four grades. Each wolf represents a candidate solution, while the most optimal solution is \( \alpha_T \), the suboptimal solution is \( \beta_T \), the third optimal solution is \( \delta_T \), and the last is \( \omega_T \). In each iteration, the three optimum solutions as \( \alpha_T, \beta_T, \delta_T \) and the suboptimal solution \( \omega_T \) are updated by Eq. (13) to promote the optimization process.

\[
\begin{align*}
&\alpha_T \leftarrow \alpha_T, \\
&\beta_T \leftarrow \frac{\alpha_T + \beta_T}{2}, \\
&\delta_T \leftarrow \frac{\alpha_T + \delta_T}{2}, \\
&\omega_T \leftarrow \frac{\alpha_T + \omega_T}{2}.
\end{align*}
\]

Figure 5: LSTM memory unit [24].

Table 3: LSTM parameters interpretation.

| Symbol   | Meaning                                                      |
|----------|--------------------------------------------------------------|
| \( c_n \) | The updated state of a memory cell                           |
| \( i_n, f_n, p_n, c_n, h_n \) | Input gate, forgetting gate, output gate, memory cell, hidden layer output value |
| \( w_{ao}, w_{ch} \) | The weight of \( o_m(n) \) and the hidden layer             |
| \( c_{n-1}, h_{n-1} \) | The output of the memory cell and the hidden layer          |
| \( w_{oi}, w_{hi}, w_{ci} \) | The weight of the input gate of \( o_m(n) \), the hidden layer, and the memory cell |
| \( w_{oj}, w_{hj}, w_{cj} \) | The weight of the forgetting gate of \( o_m(n) \), the hidden layer, and the memory cell |
| \( w_{op}, w_{hp}, w_{cp} \) | The weight of the output gate of \( o_m(n) \), the hidden layer, and the memory cell |
| \( b_i, b_f, b_o, b_p \) | Activation function, Offset item                            |
| \( n \times n \) | Dot product                                                  |

where \( \bar{c}_n \) denotes weight and \( D_0^p \) denotes prediction. LSTM-KNN pseudocode is shown in Algorithm 2.

The number of units in the hidden layer is \( n_{\text{unit}} \), and the data dimension \( D \) of LSTM in each thread is 1, so the time complexity is \( 4 \times (n_{\text{unit}}^2 + 2n_{\text{unit}}) \). On account of the parallel calculation structure, the total time does not change too much with the increasing number of threads. Compared with the original complexity \( 4 \times (n_{\text{unit}}^2 + n_{\text{unit}} \times D + n_{\text{unit}}) \), which reduces a lot of computation and improves efficiency. The time complexity of KNN is \( O(n) \), which is only related to the number of data, so the calculation speed is fast.

LSTM has good robustness in traffic prediction [60] and also improves performance by setting reasonable time lag [61] or forgetting layer. KNN can avoid large deviation directly because it calculates the closest Euclidean distance to produce the prediction [62]. These two positive aspects of robustness will support the proposed method to gain more adaptability on data fluctuations and environmental change.
determine the prey position and direct the $\omega_t$ to update the position around it [63].

By using the improved adaptive convergence factor [64], the extremum can be quickly found when the step size is large in a global search. Besides, the extremum can be prevented from missing when the step size becomes smaller in the local search. The weight step size formula [64] adds the weight decreasing strategy, which can reduce unnecessary iterative processes and improve efficiency. The calculation method is as follows.

(1) Initialize the Population. The upper bound $U_b$ and lower bound $L_b$ are defined, respectively. The number of wolves is $N$. The dimensions are $S$. $M_{N \times 5}$ denotes an $N \times 5$ two-dimensional matrix, which is the searching field. There are $2m + 1$ key parameters to be optimized in each element of the field array, which are step, epochs in the LSTMs module, and the $K$ value in the KNN module. Generate integers randomly at their respective upper and lower bounds to form an element of field array:

$$M_{[i,j]} = [\text{step}_1, \text{step}_2, \ldots, \text{step}_n, \text{epochs}_1, \text{epochs}_2, \ldots, \text{epochs}_n, k].$$

(14)

Each parameter has a different $U_b$ and $L_b$. Set up two vectors to record the bounds.

$$U_{\text{TGWO}} = [U_1, U_2, \ldots, U_{2m+1}], L_{\text{TGWO}} = [L_1, L_2, \ldots, L_{2m+1}].$$

(15)

(2) Calculate Fitness. Input the element corresponding to each wolf into LSTM-KNN and compare the error. Define the optimal solution as $\alpha_T$, $\beta_T$, $\delta_T$, respectively.

(3) Update location with $\alpha_T$, $\beta_T$, $\delta_T$:

$$D_A = \left| C_T \cdot W_a(t) - W(t) \right|,$$

$$D_B = \left| C_T \cdot W_B(t) - W(t) \right|,$$

$$D_\delta = \left| C_T \cdot W_\delta(t) - W(t) \right|,$$

$$W_a(t + 1) = \varphi W_a(t) - A_T \cdot D_A,$$

$$W_B(t + 1) = \varphi W_B(t) - A_T \cdot D_B,$$

$$W_\delta(t + 1) = W_a(t + 1) + W_B(t + 1) + W_\delta(t + 1),$$

$$A_T = 2a_T (r_1 - 1),$$

$$C_T = 2r_2,$$

$$a_T = a_{\text{max}} - \left[ \frac{1 + \exp \left( -\frac{t}{t_{\text{max}}} \right)}{2} \right],$$

(16)

where $D$ denotes the distance between the grey wolves and their prey, $t$ denotes current iteration times, $W_p(t)$ denotes the position of the grey wolf in $t$ iteration times, $W(t)$ denotes the position of the prey in $t$ iteration times, $A$ and $C$ denote the coefficient vectors, $r_1$ and $r_2$ denote random coefficients with scalars between 0 and 1, generally take 0.5, $a$ denotes the convergence factor, $\varphi$ denotes the weight of
3. Experiments and Analysis

3.1. Experimental Data. Whitemud Drive is an in-city highway across Edmonton, Alberta, Canada. It is 28 kilometers long with a basic speed limit of 80 kilometers per hour. As a test road, Whitemud Drive is equipped with seven traffic video cameras and seven loop detectors (VDS1017, VDS1037, VDS1034, VDS1031, VDS1029, VDS1027, and VDS1019) from west to east on the main road and gate road to observe the vehicle flow, the vehicle speed, and the vehicle density. In this paper, data of 15 working days are used as historical data for experiments.

3.2. GMM Selecting Test. VDS1019 is set as the prediction point $D_0$, and the change of traffic flow within one day of the working day is plotted according to 5 minutes (see Figure 6). To better carry out the time-division work, the data are smoothed by SG, and the image is reconstructed (see Figure 7).

Find the extreme values, set up the midpoints, and divide the time series (see Figure 8).

The time-division results are shown in Table 4.

3.2.1. Reconstructing the Dataset by Time-Division. VDS1017, VDS1037, VDS1034, VDS1031, VDS1029, and VDS1027 are recorded as $O_1 \sim O_6$, their historical data on working days are divided into parts according to $T_1 \sim T_4$, and the data in the same time part are put into the same column. Since the length of the road is 28 km and the speed limit is 80 km/h, we choose 60 km/h as the test speed. It takes 30 minutes to go from VDS1017 to VDS1019. Considering the continuity of the road network, the vehicle passes through each point every 5 minutes, so $O_2 \sim O_5$ is delayed by 5–25 minutes for input. And the prediction point $D_0$ is delayed by 30 minutes for input. In this way, the data of each day are the delayed input to form a dataset.

Calculate KL divergence and GRA coefficient (see Table 5).

Input the result table into the GMM module for classification.

Taking $T_2$ as an example, the GMM classification results are shown in Figure 9.

The classification results are two types. The closer the GRA coefficient is to 1, the better the correlation is, and the closer the KL divergence is to 0, the more similar the distribution is. So, choose the yellow mark points as the input points (see Figure 9). The following is the final classification results of four datasets (see Table 6).

3.3. Comparative Experiments at Different Upstream Points. Different upstream points are selected for model prediction and mean absolute percentage error (MAPE) comparison so as to verify the effect after classification. Results are shown in Table 7.

In this dataset, it can be seen that the abandoned upstream points have little difference from the selected points (see Table 5). So it is not obvious in the accuracy improvement, which is reasonable. If in datasets with a large difference, the meaning of GMM classification operation will be reflected more.

3.4. TGWO-LSTM-KNN Experiments

3.4.1. LSTM-KNN Structural Test. Considering the operation time and efficiency, this paper constructs the following structures for testing. Since the training data are 15 days’ traffic flow, the batch size is set to 15, and the data are divided into 15 groups, which will ignore the data relationship between each group and reduce the risk of overfitting. If the batch size is too small, it is not conducive to the training model and is easy to overfit. The test step is 3, epochs are 100, and the comparison results are shown in Table 8.

The results show that increasing the number of LSTM layers can improve the prediction accuracy, but it increases a lot of computing time and overfits easily. Increasing the forgetting layer (the forgetting rate is 0.2) will reduce the accuracy. In the first-layer structure, there is only a small difference between 256 units and 128 units; therefore, the single-layer 128 units structure is selected for prediction in the following LSTM.

3.4.2. LSTM-KNN Time Steps Test. The model is tested under different time steps (5 minutes, 10 minutes, 15 minutes, and 30 minutes). The result shows that the model has good accuracy and stability (see Figure 10). The overall prediction accuracy shows a downward trend, and the absolute error shows an upward trend. It is worth noting that when the time step is 10 minutes, the trend of accuracy will change and the error is lower than 15 minutes. In general, even if the time step is different, the model still has good performance for the prediction accuracy.
3.4.3. TGWO Optimization Test. The dataset has four upstream points, and the key parameters to be optimized are the steps of $O_2, O_4, O_5, O_6$, the training epochs in the LSTM module, and the K in the KNN module. So, $M_{[i,j]} = [\text{step}_2, \text{step}_4, \text{step}_5, \text{step}_6, \text{epochs}_2, \text{epochs}_4, \text{epochs}_5, \text{epochs}_6, \text{K}]$. Set the upper and lower bounds: the step size ranges from 3 to 20. Training time ranges from 100 to 200. The coefficient K ranges from 5 to 50, the number of iterations $t$ is 50, the number of wolves is $N = 5$, and the dimension $S$ is 30. The training results are shown in Figure 11.

When the MAPE is 10.04 (green mark in Figure 11), it will reach the optimal solution $M_{\text{best}} = [5, 6, 5, 6, 120, 100, 120, 100, 25]$. The steps of $O_2, O_5$ and $O_6$ are 5 and 6, the training epochs are 120 and 100, and K is 25.

### Table 4: Time-division results.

| Time part name | Number of data | Time      |
|----------------|----------------|-----------|
| $T_1$          | 0–70           | 0:00–7:00 |
| $T_2$          | 71–110         | 7:00–9:00 |
| $T_3$          | 111–155        | 9:00–13:00|
| $T_4$          | 156–249        | 13:00–20:00|

### Table 5: Calculation results of KL divergence and GRA coefficient.

| KL/GRA result | T1 | T2 | T3 | T4 |
|---------------|----|----|----|----|
| KL            | 0.030 | 0.029 | 0.038 | 0.035 |
| GRA           | 0.90 | 0.90 | 0.89 | 0.90 |
| KL            | 0.023 | 0.009 | 0.021 | 0.009 |
| GRA           | 0.78 | 0.81 | 0.78 | 0.81 |
| KL            | 0.020 | 0.017 | 0.031 | 0.018 |
| GRA           | 0.86 | 0.88 | 0.85 | 0.88 |
| KL            | 0.010 | 0.017 | 0.035 | 0.018 |
| GRA           | 0.87 | 0.88 | 0.82 | 0.88 |
| KL            | 0.015 | 0.009 | 0.031 | 0.018 |
| GRA           | 0.82 | 0.89 | 0.85 | 0.89 |
| KL            | 0.004 | 0.004 | 0.004 | 0.004 |
| GRA           | 0.92 | 0.92 | 0.92 | 0.92 |

### Table 6: Classification results of GMM.

| GMM result | T1 | T2 | T3 | T4 |
|------------|----|----|----|----|
| D point    | $O_6, O_4, O_5, O_6$ | $O_2, O_5, O_6, O_6$ | $O_1, O_2, O_3, O_5, O_6$ | $O_1, O_2, O_3, O_6$ |

20. Training time ranges from 100 to 200. The coefficient K ranges from 5 to 50, the number of iterations $t$ is 50, the number of wolves is $N = 5$, and the dimension $S$ is 30. The training results are shown in Figure 11.

When the MAPE is 10.04 (green mark in Figure 11), it will reach the optimal solution $M_{\text{best}} = [5, 6, 5, 6, 120, 100, 120, 100, 25]$. The steps of $O_2, O_5$ and $O_6$ are 5 and 6, the training epochs are 120 and 100, and K is 25.

### 3.4.4. Comparison of TGWO-LSTM-KNN and Other Models.

TGWO-LSTM-KNN is compared with SVM, LSTM, and BP. The parameters of TGWO-LSTM-KNN are the optimal
solution. The step of LSTM-KNN is 3, K is 16, and the epochs is 100. LSTM has 256 units in the first layer and 128 units in the second layer of the double-hidden layer structure, with the step of 3 and epochs of 100. SVM uses three modes to fit, linear and poly, RBF. BP neural network is a three-layer structure, two fully connected layers, the middle increased the forgetting layer (rate is 0.2), and the epochs is 100. Results are shown in Figure 12.

The accuracy of LSTM-KNN can reach the level of the popular model. The accuracy of TGWO-LSTM-KNN can be improved by 15.27% compared with single-LSTM, 9.47% compared with BP, and 43.12% compared with poly-SVM (see Table 9). Besides, the advantage of the hybrid model is not accuracy, but being able to output the prediction set of upstream points for collaborative management.

---

**Table 7: MAPE comparison table of different points.**

| Model/points | V34    | V34 + V31 | V34 + V31 + V29 | V34 + V31 + V29 + V27 | Including abandonment points |
|--------------|--------|-----------|------------------|------------------------|----------------------------|
| LSTM-KNN     | 14.45  | 14.96     | 14.31            | 12.19                  | 12.46                      |
| Linear-SVM   | 20.47  | 21.14     | 20.03            | 21.17                  | 21.24                      |
| Poly-SVM     | 15.85  | 17.30     | 17.15            | 17.65                  | 18.04                      |
| RBF-SVM      | 20.51  | 20.29     | 20.87            | 20.17                  | 20.48                      |
| Multi-LSTM   | 14.32  | 14.51     | 11.96            | 9.51                   | 23.14                      |

---

**Table 8: Error comparison table of different structures.**

| Structure                             | RMSE   | MAE   | MAPE (%) |
|---------------------------------------|--------|-------|----------|
| Double-layer structure                |        |       |          |
| LSTM (256 units) + LSTM (128 units)   | 42.49  | 29.81 | 11.64    |
| Four-layer structure                  |        |       |          |
| LSTM (256 units) + Dropout (0.2) + LSTM (128 units) + Dropout (0.2) | 44.87  | 31.33 | 12.25    |
| Single-layer structure                |        |       |          |
| LSTM (256 units)                      | 44.42  | 31.55 | 12.32    |
| Single-layer structure                |        |       |          |
| LSTM (128 units)                      | 44.43  | 31.29 | 12.19    |

---

**Figure 10: Different error (MAPE, MAE, RMSE) comparisons of different time steps.**
4. Conclusions

In this paper, the TGWO-LSTM-KNN prediction model with GMM classification considering spatial-temporal characteristics under the concept of attention mechanism is proposed. The time series is divided into parts by using the temporal characteristic of the prediction point. And GMM through KL and GRA is used for further classification. The upstream points with a small difference in distribution and high linear similarity are selected to increase the accuracy and reduce the complexity. Then the hybrid model TGWO-LSTM-KNN is used to train and predict. Parallel computing is used in the LSTM module to improve efficiency.
GMM as an unsupervised model can be very flexible to classify. This model can be applied to the multifactor model to reduce complexity. KNN, as the next part of LSTM, fully combines upstream points data and prediction points data for prediction. Compared with SVM, KNN has the characteristics of fast speed and greater data processing ability, which is more suitable for multiple factors and complex data of freeway. LSTMs module can ignore the relationship between upstream points to perform the parallel computation. It can reduce the operation time and make the model more practical. TGWO is less likely to fall into local station. It can reduce the operation time and make the data off freeway. LSTM’s module can ignore the relationship which is more suitable for multiple factors and complex characteristics of fast speed and greater data processing ability, for prediction. Compared with SVM, KNN has the characteristic of low computation complexity. Therefore, it can be applied in real freeways to achieve the purpose of collaborative management.

Data Availability

The data used to support the findings of this study are openly available in the OpenITS platform for noncommercial purposes only and the website is https://www.openits.cn/openData1/700.jhtml.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors want to thank Tianjiao Wang, Xi Chen, Xiang Wang, and Lingbin Kong for their help in the study. Thanks also should be given to the OpenITS platform (http://www.openits.cn/). The research and publication of this article were funded by the National Natural Science Foundation of China (52072129).

References

[1] C. Qi and Z.-S. Hou, "Application of adaptive single-exponent smoothing for short-term traffic flow prediction," Control Theory & Applications, vol. 29, no. 4, pp. 465–469, 2012.
[2] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg-marquardt algorithm," IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 2, pp. 644–654, 2012.
[3] H. Gao and D. Zhang, "Traffic flow forecasting model based on fractal and three-exponential smoothing," Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition), vol. 38, no. 6, pp. 63–67, 2018.
[4] L. Lv, M. Chen, Y. Liu, and X. Yu, "A plane moving average algorithm for short-term traffic flow prediction," in Advances in Knowledge Discovery and Data Mining, T. Cao, E. P. Lim, Z. H. Zhou, T. B. Ho, D. Cheung, and H. Motoda, Eds., pp. 357–369, Springer, Cham, Switzerland, 2015.
[5] T. Mai, B. Ghosh, and S. Wilson, "Short-term traffic-flow forecasting with auto-regressive moving average models," Proceedings of the Institution of Civil Engineers - Transport, vol. 167, no. 4, pp. 232–239, 2014.
[6] D. Billings and Y. Jian-Shiou, "Application of the ARIMA models to urban roadway travel time prediction - a case study," in Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics, vol. 1-6, p. 2529, Toronto, ON, Canada, October 2006.
[7] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," European Transport Research Review, vol. 7, no. 3, 2015.
[8] J. Liu and W. Guan, "A summary of traffic flow forecasting methods," Journal of Highway and Transportation Research and Development, vol. 3, pp. 82–85, 2004.
[9] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," Transportation Research Part C: Emerging Technologies, vol. 43, pp. 50–64, 2014.
[10] L. L. Ojeda, A. Y. Kibangou, and C. C. De Wit, "Adaptive kalman filtering for multi-step ahead traffic flow prediction," in Proceedings of the 2013 American Control Conference, pp. 4724–4729, Washington, DC, USA, June 2013.
[11] Z. Liyan, M. Jian, and S. Jian, "Examples of validating an adaptive kalman filter model for short-term traffic flow prediction," in Proceedings of the Twelfth COTA International Conference of Transportation Professionals, Xi’an, China, December 2012.
[12] T. Zhou, D. Jiang, Z. Lin, G. Han, X. Xu, and J. Qin, "Hybrid dual Kalman filtering model for short-term traffic flow forecasting," IET Intelligent Transport Systems, vol. 13, no. 6, pp. 1023–1032, 2019.
[13] S. Zhang, Y. Song, D. Jiang, T. Zhou, and J. Qin, "Noise-identified kalman filter for short-term traffic flow forecasting," in Proceedings of the 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), Shenzhen, China, December 2019.
[14] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved K-nearest neighbor model for short-term traffic flow prediction," Procedia - Social and Behavioral Sciences, vol. 96, pp. 653–662, 2013.
[15] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," Journal of Transportation Engineering, vol. 132, no. 2, pp. 114–121, 2006.
[16] Y. Zhang and Y. Xie, "Forecasting of short-term freeway volume with v-support vector machines," Transportation Research Record: Journal of the Transportation Research Board, vol. 2024, no. 1, pp. 92–99, 2007.
[17] Y. Sun, B. Leng, and W. Guan, "A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system," Neurocomputing, vol. 166, pp. 109–121, 2015.
[18] D. Min, "Short-time prediction of traffic flow based on PSO optimized SVM," in Proceedings of the 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xiamen, China, January 2018.
[19] B. L. Smith and M. J. Demetsky, "Short-term traffic flow prediction models—a comparison of neural network and nonparametric regression approaches," in Proceedings of the Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Bari, Italy, October 1994.
[20] S. Dehuri and S.-B. Cho, "A comprehensive survey on functional link neural networks and an adaptive PSO-BP learning for CFLNN," Neural Computing & Applications, vol. 19, no. 2, pp. 187–205, 2010.
[21] S. Wang, N. Zhang, L. Wu, and Y. Wang, “Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method,” Renewable Energy, vol. 94, pp. 629–636, 2016.

[22] P. Vincent, H. Larochele, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion,” Journal of Machine Learning Research, vol. 11, pp. 3371–3408, 2010.

[23] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: continual prediction with LSTM,” Neural Computation, vol. 12, no. 10, pp. 2451–2471, 2000.

[24] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, “Learning precise timing with LSTM recurrent networks,” Journal of Machine Learning Research, vol. 3, no. 1, pp. 115–143, 2003.

[26] M. Sundermeyer, R. Schluter, and H. Ney, LSTM Neural Networks for Language Modeling, Interspeech 2012, Portland, OR, USA, 2012.

[27] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, “A C-LSTM neural network for text classification,” Computer Science, vol. 1, no. 4, pp. 39–44, 2015.

[28] A. Graves and J. Schmidhuber, “Framework phoneme classification with bidirectional LSTM and other neural network architectures,” Neural Networks, vol. 18, no. 5–6, pp. 602–610, 2005.

[29] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” in Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, November 2016.

[30] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-term residential load forecasting based on LSTM recurrent neural network,” Ieee Transactions on Smart Grid, vol. 10, no. 1, pp. 841–851, 2019.

[31] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: human trajectory prediction in crowded spaces,” in Proceedings of the 2016 Ieee Conference on Computer Vision and Pattern Recognition, pp. 961–971, Las Vegas, NV, USA, June 2016.

[32] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale Acoustic modeling,” in Proceedings of the 15th Annual Conference of the International Speech Communication Association, pp. 338–342, Graz, Austria, September 2014.

[33] H. Wen, D. Zhang, and L. Suiyan, “Application of GA-LSTM model in highway traffic flow prediction,” Journal of Harbin Institute of Technology, vol. 51, no. 9, pp. 81–87+95, 2019.

[34] J. V. Tu, “Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes,” Journal of Clinical Epidemiology, vol. 49, no. 11, pp. 1225–1231, 1996.

[35] J. Guo, Y. Liu, and L. Ma, Assignee. EEMD-IAGA-BP Neural Network Based Ship Traffic Flow Predicting Method, Involves Constructing Three-Layer BP Neural Network with Enhanced Adaptive Genetic Algorithm Optimization as Training Model, and Obtaining Predicted Results Patent CN110111606-A, Inventors; Univ Shanghai Maritime, Shanghai Shi, China, 2021.

[36] S. Y. Liang and C. Sivapragasam, “Flood stage forecasting with support vector machines,” Journal of the American Water Resources Association, vol. 38, no. 1, pp. 173–186, 2002.

[37] M. Duo, Y. Qi, G. Lina, and E. Xu, “A short-term traffic flow prediction model based on EMD and GPSO-SVM,” in Proceedings of the 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, March 2017.

[38] H. Chung and K.-s andShin, “Genetic algorithm-optimized long short-term memory network for stock market prediction,” Sustainability, vol. 10, no. 10, 2018.

[39] W. Zhou and S. Meng, Assignee. Navigation Reminder Method Based on SVD-PSO-LSTM for Predicting Short-Term Traffic Flow, Involves Providing First Train and Optimizes LSTM Model, Then Collects Historical Traffic Flow Data, and Preprocesses and Input to Training Flow Data Patent CN111709549-A, Inventors; Univ Donghua, Shanghai, China, 2021.

[40] X. Luo, D. Li, Y. Yang, and S. Zhang, “Spatiotemporal traffic flow prediction with KNN and LSTM,” Journal of Advanced Transportation, vol. 2019, 10 pages, 2019.

[41] L. Dai, H. Mei, C. Qian, M. Yun, and L. Jin-Ming, “Survey on short-term traffic flow forecasting based on deep learning,” Computer Science, vol. 46, no. 3, pp. 39–47, 2019.

[42] F. Su, H. Dong, L. Jia, Z. Tian, and X. Sun, “Space-time correlation analysis of traffic flow on road network,” International Journal of Modern Physics B, vol. 31, no. 5, 2017.

[43] Q. Liu, Y. Cai, H. Jiang, X. Chen, and J. Lu, “Traffic state spatial-temporal characteristic analysis and short-term forecasting based on manifold similarity,” Ieee Access, vol. 6, pp. 9690–9702, 2018.

[44] C. Wang, “Attention-based traffic flow prediction and research,” East China Jiaotong University, Nanchang, China, Degree Diss, 2020.

[45] L. Li, “Research on spatial temporal prediction model of traffic flow based on attentional mechanism,” South China University of Technology, Guangzhou, China, Degree Diss, 2020.

[46] M. Chen, “Research on traffic flow prediction based on improved graph attention network,” Shanghai Normal University, Shanghai, China, Degree Diss, 2021.

[47] D. Ma, X. B. Song, J. Zhu, and W. Ma, “Input data selection for daily traffic flow forecasting through contextual mining and intra-day pattern recognition,” Expert Systems with Applications, vol. 176, 2021.

[48] D. Ma, X. Song, and P. Li, “Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter- and intra-day traffic patterns,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 5, pp. 2627–2636, 2021.

[49] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, “Daily long-term traffic flow forecasting based on a deep neural network,” Expert Systems with Applications, vol. 121, pp. 304–312, 2019.

[50] X. Song, W. Li, D. Ma, D. Wang, L. Qu, and Y. Wang, “A match-then-predict method for daily traffic flow forecasting based on group method of data handling,” Computer-Aided Civil and Infrastructure Engineering, vol. 33, no. 11, pp. 982–998, 2018.

[51] D. Ma, B. Sheng, X. Ma, and S. Jin, “Fuzzy hybrid framework with dynamic weights for short-term traffic flow prediction by mining spatio-temporal correlations,” IET Intelligent Transport Systems, vol. 14, no. 2, pp. 73–81, 2020.

[52] S. Hao, D.-H. Lee, and D. Zhao, “Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system,” Transportation Research Part C: Emerging Technologies, vol. 107, pp. 287–300, 2019.

[53] J. Steiner, Y. Termonia, and J. Deltour, “Smoothing and differentiation of data by simplified least square procedure,” Analytical Chemistry, vol. 44, no. 11, pp. 1906–1909, 1972.
[54] D. Reynolds, *Gaussian Mixture Models*, Springer US, Berlin, Germany, 2008.

[55] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[56] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[57] S. Liu, Z. Wen, J. Tao et al., "A data driven method for target and concatenation cost calculation with KL-divergence in Mandarin hybrid speech synthesis," in *Proceedings of the 2014 12th International Conference on Signal Processing*, pp. 572–576, HangZhou, China, October 2014.

[58] N. Gerhard, *Pearson Correlation Coefficient*, p. 132, Springer, Vienna, Austria, 2009.

[59] J. Deng, "Introduction to grey system theory," *Journal of Grey System*, vol. 1, no. 1, pp. 1–24, 1989.

[60] X. Wang, L. Xu, and K. Chen, "Data-driven short-term forecasting for urban road network traffic based on data processing and LSTM-RNN," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3043–3060, 2019.

[61] M. Sangiorgio and F. Dercole, "Robustness of LSTM neural networks for multi-step forecasting of chaotic time series," *Chaos Solitons & Fractals*, vol. 139, 2020.

[62] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 61–78, 2016.

[63] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[64] W. S. Zhang, Z. Q Hao, J. J. Zhu, T. T. Du, and H. M. Hao, "BP neural network model for short-time traffic flow forecasting based on transformed grey wolf optimizer algorithm," *Journal of Transportation Systems Engineering & Information Technology*, vol. 20, no. 02, pp. 196–203, 2020.