

RESEARCH ARTICLE

Combined Prediction of Short-Term Travel Time of Expressway Based on CEEMDAN Decomposition

XINGLI JIA¹, WUXIAO ZHOU², SHUANGQING LI¹, AND XINGPENG CHEN¹¹School of Highway, Chang'an University, Xi'an 710064, China²Jiangsu Shagang Logistics Transportation Management Company Ltd., Suzhou 715000, China

Corresponding author: Xingli Jia (jiaxingli@chd.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFC1512003; in part by the Fundamental Research Funds for the Central Universities, Chang'an University (CHD) under Grant 300102212203; and in part by the Natural Science Foundation of Shaanxi Province under Grant 2020JM-260.


ABSTRACT Travel time is the basis for intelligent emergency control and guidance in expressway networks. To realize its accurate prediction and improve the expressway service level during emergencies, this study uses a combined model to predict the short-term travel time of expressway sections based on the expressway gantry data of Sichuan Province. First, the travel time series was extracted using a data matching algorithm, and the double standard deviation-cyclic elimination (2SD-CE) algorithm was used to clean the data. Then, combined with the complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) algorithm, the travel time subsequence was extracted, and the frequency of the subsequence was divided by Sample entropy (SampEn) algorithm. Based on this, bidirectional long short-term memory (BiLSTM), long short-term memory (LSTM), and vanilla recurrent neural network (vanilla RNN) models were used to construct prediction combination model 1 (CM1) under the condition of a single feature. Subsequently, the CEEMDAN and empirical mode decomposition (EMD) algorithms were combined with the LSTM algorithm to obtain the combination models (CM2 and CM3) without frequency division. The example calculation and analysis show that under different time granularities (5 min, 10 min, and 15 min) and different highway sections, the combined model can integrate the advantages of all prediction models and has higher prediction accuracy and stability, among which the prediction effect of CM1 can reduce the prediction value of the root mean squared error (RMSE) by 18.8~26.4%, 0.8~41%, 4.1~13.3%.

INDEX TERMS Expressway, travel time prediction, deep learning, CEEMDAN, recurrent neural network.

I. INTRODUCTION

After scattered or local traffic accidents, natural disasters, and other emergencies in the road network have a congestion impact on the expressway, this impact spreads to the regional road network and slows down the efficiency of emergency rescue. Therefore, there is an urgent need to improve the intelligent control and guidance ability of expressway networks, strengthen the close cooperation between people, vehicles, and roads, improve road traffic efficiency, and

create an efficient, accurate, and real-time expressway operation system [1]. In recent years, with the continuous development of big data and artificial intelligence technology, traffic data collected by sensors have gradually improved, providing a certain data basis for the construction of machine learning models. As a new research direction in machine learning, deep learning has also been increasingly applied in the field of traffic prediction [2], [3], [4], [5]. Deep learning can not only learn its internal laws and high-order representation from massive traffic data but also has an end-to-end learning method that is suitable for short-term travel time prediction problems with high nonlinearity [6]. Accurate and real-time

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamed M. A. Moustafa .

travel time prediction is the key link between intelligent guidance and control of expressway networks. Improving the prediction accuracy of travel time can not only provide detection means for emergencies, but also improve the traffic efficiency of road networks through timely regulation and control to achieve smooth road networks.

Travel time prediction is the basis for the intelligent control and guidance of expressway networks. As an objective reflection of the driving condition of vehicles in a certain section, it can measure the traffic state of the section at different times and is closely related to traffic parameters such as traffic flow, driving speed, and time occupancy [7]. Travel time prediction can be divided into long-, medium-, and short-term predictions. Medium- and long-term predictions, as important considerations for the long-term planning of road networks, usually take years and months as scale units. The short-term travel time prediction has a smaller scale than the medium- and long-term predictions, but its accuracy is higher. It is generally believed that a prediction with a time span of less than or equal to 15 min is a short-term prediction [9], [10], [11].

Real-time and accurate short-term travel time prediction can correctly capture the change law of traffic flow on the road to reasonably infer the traffic state of the outlet network or road section at the next moment. In the decades of research and development of traffic prediction, scholars at home and abroad have developed various algorithms to predict traffic parameters such as traffic flow, speed, and travel time. According to different prediction methods, travel time prediction models are divided into two categories: statistical models, which can be divided into linear and nonlinear theoretical statistical models. Linear theoretical models include Kalman filtering and time-series methods. Kalman filtering theory was proposed in the 1960s, and Okutani and Stephanedes [12] first applied this theory to traffic flow prediction. The Kalman filtering model can deal with stationary or non-stationary data, but it is a linear model, and the actual traffic prediction problems are mostly nonlinear problems; therefore, the Kalman filtering model has some limitations in practical applications. The autoregressive moving average model (ARMA) and autoregressive integrated moving average model (ARIMA) are commonly used time series prediction models [13]. Ahmaed and Cook [14] applied the ARIMA model for traffic flow prediction for the first time. Li *et al.* [15] predicted traffic flow based on an improved ARIMA model. These types of time series models are simple in modeling, but they have high requirements for continuity of data and it is difficult to deal with complex prediction problems of multi-dimensional inputs.

Nonlinear theoretical models include nonparametric regression [16] and chaos theory [17]. The k-nearest neighbor (KNN) algorithm is a typical non-parametric regression model. On the one hand, its algorithm is simple and easy to understand, on the other hand, it relies heavily on training data and has poor fault tolerance to training data. Disbro and Frame [18] introduced the chaos theory into the field of transportation for the first time. Wang and Shi [19] built

a nonlinear chaos prediction model based on phase-space reconstruction theory to predict urban road traffic flow. The chaos theory model is based on measured data to obtain the chaotic characteristic parameters of the system, which avoids the influence of subjective factors and has a high prediction accuracy, but it is only suitable for short-term traffic flow prediction.

The second category is artificial intelligence (AI) technology based on neural networks. In recent years, with the rapid development of artificial intelligence, deep learning theory and neural network models have provided additional modeling ideas for the study of traffic flow prediction. Wu *et al.* [20] applied SVR to travel time prediction and compared it with historical mean and other methods, and the results showed that the model could significantly reduce the prediction error. Su *et al.* [21] proposed a short-term traffic flow prediction method based on incremental support vector regression (ISVR), and the results showed that the prediction accuracy of this method was better than that of BP neural network model. Luo *et al.* [22] used the least square support vector machine method to predict the traffic flow, and adopted the fusion optimization algorithm to select the optimal parameters, which improved the prediction ability and calculation efficiency of the model. Although these traditional neural networks can better learn the characteristics of traffic flow and predict future traffic flow according to the temporal and spatial variation characteristics of traffic flow, most of them use single hidden layer networks, which cannot learn the deeper variation characteristics of traffic flow data, and the prediction accuracy is often lower than that of deep network prediction methods.

Deep learning has a strong learning ability for time series and can deal better with spatially or temporally related data structures [23]. The depth of a recurrent neural network (RNN) is not only reflected in the fact that it has multiple hidden layer structures but also has the function of time memory. RNN can be used for the recognition of text, speech, and other data sequences, and can be better applied to the relevant prediction field of time series data [24]. Serious gradient vanishing and gradient explosion problems exist in RNN. Subsequently, Hochreiter and Schmidhuber [25] designed a long short-term memory (LSTM) unit to overcome this defect, which enabled the recurrent neural network represented by LSTM to be applied on a large scale in the field of time-series prediction. Ma *et al.* [26] used the LSTM structure to establish a traffic speed prediction model, and used the microwave traffic speed data of Beijing for verification. The experimental results showed that the network effectively captured the correlation and nonlinearity of the traffic state time, and the prediction accuracy was better than that of most statistical methods.

In conclusion, the deep learning model has a better prediction effect than the traditional traffic prediction methods. Therefore, in the face of a large amount of diversified traffic data, selecting the appropriate model or combining models with different structures to realize the complementary

advantages of the model, extracting significant traffic features, and setting appropriate parameters to further improve the accuracy of expressway short-term prediction is the development direction of deep learning in the field of expressway short-term travel time prediction in the future. The main contributions of this study are as follows.

(1) A short-term travel-time prediction model based on complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) was proposed. In view of the high nonlinearity of travel time, the CEEMDAN algorithm is used to decompose the travel time series.

(2) Combined with the Sample entropy (SampEn) algorithm, the complexity of each subsequence after decomposition was calculated. According to the complexity of time-series components, they are divided into high-frequency, intermediate frequency, and low-frequency sequences, which are predicted by the bidirectional long short-term memory (BiLSTM), LSTM, and vanilla RNN models, respectively.

(3) The prediction results of each high-frequency, medium-frequency, and low-frequency sequence component are superimposed to obtain the travel time prediction value, which is compared with the baseline models. The research results can provide a basis for the intelligent control and guidance of expressway networks.

The rest of the paper is organized as follows: The section II is related works. In section III, a method that combines the CEEMDAN, SampEn, and LSTM was proposed. The section IV describes the dataset of this study, and uses a novel algorithm to complete the preprocessing of the dataset. The V part is the validation and evaluation of the proposed model through experiments on different time granularity and different expressways. Finally, conclusions are drawn in section VI.

II. RELATED WORKS

This section briefly reviews some methods in the literature for extracting more significant traffic flow features through a combination of optimization algorithms and models. Because traffic flow data are disturbed by various factors such as weather and traffic detectors during the collection process, they often contain a large amount of noise. Currently, a large number of studies have shown that decomposition algorithms can reduce the influence of noise on prediction models and improve prediction accuracy. The empirical modal decomposition (EMD) method, proposed by Huang *et al.* [27], is a processing method that can cope with nonlinear sequences. EMD can be used without setting arbitrary basis functions and can be analyzed directly according to the data scales and characteristics. It has been shown to be effective when applied to decomposing highly nonlinear and nonsmooth data [28]. Duo *et al.* [29] used EMD to decompose traffic flow sequences into different frequency components and then input them into an optimized SVM model, which was validated by a dataset of the Changchun city road network, showing that EMD can achieve improved prediction accuracy. Du *et al.* [30] proposed a prediction model based on empirical

mode decomposition (EMD) and gated recurrent unit (GRU) neural network for a more comprehensive characterization of network traffic, by EMD to the traffic data is decomposed into multiple components, and each component is used to train the corresponding GRU neural network, and finally, the predicted values of all components are combined to obtain the final result. Although the EMD algorithm can effectively cope with nonlinear sequences, the mode aliasing occurs during the decomposition process, which is also a limitation of the EMD algorithm. EEMD [31] solved the problem of mode mixing by additional white noise. Tang *et al.* [32] compared five denoising schemes and proposed that EEMD is superior to other algorithms. Liu *et al.* [33] used the EEMD algorithm to decompose the time series and extracted the basic feature subset of each component using the minimum redundancy maximum association feature selection algorithm, and then used deep belief network to each component is trained, and finally the prediction results are aggregated into the output of the integrated model, and the results show that the method has significant performance improvement compared with a single deep belief network and other selected methods. Based on this, Torres *et al.* [34] proposed the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) algorithm [28] by adding adaptive white noise to each decomposition in order to improve the completeness of EEMD and reduce reconstruction errors. The basic principle is to adaptively add white noise during intrinsic mode function (IMF) component decomposition to calculate each intrinsic mode functions (IMFs) component, which can achieve almost zero reconstruction error, has good integrity, reduces the number of integrations, and alleviates the phenomenon of modal aliasing. Guo *et al.* [35] proposed a hybrid model based on deep learning methods and CEMMDAN, which has great potential for traffic flow prediction. Lu *et al.* [36] used this method to decompose raw traffic flow data into several intrinsic modal function components and one residual component, and then the XGBoost model is trained and the decomposed components are predicted separately. The final prediction results were obtained by integrating the prediction outputs of the XGBoost method. It was demonstrated that the CEEMDAN-XGBoost model can effectively fit the complex fluctuations of different types of road sections, and the model accuracy is better than that of LSTM and other XGBoost-based models. In the study of Zhu *et al.* [37], the replacement entropy values of the IMF components were calculated using the PE algorithm after decomposing the original traffic flow data into several relatively stable modal components, and the components with similar entropy values were superimposed to form a new sequence. Huang *et al.* [8] evaluated five decomposition algorithms, EMD, EEMD, CEEMDAN, WPD (Wavelet Packet Decomposition), and VMD (Variational Mode Decomposition), based on BiLSTM in terms of prediction performance, robustness, and generalization performance, to investigate the impact of multi-scale decomposition algorithms on neural network models, and they concluded that CEEMDAN

can be used to pursue prediction accuracy and anti-noise performance.

From the discussion in the previous section, we learned that deep learning models have better prediction results than traditional traffic prediction methods, but that is not absolute. Single prediction models are often designed by considering the traffic flow time series only as a single series without designing the optimal method based on its inherent characteristics. Yu *et al.* first decompose and reconstruct the traffic flow data series using wavelet transform, build a sub-prediction model of the reconstructed data using radial basis function (RBF), and optimize it using Particle Swarm Optimization (PSO) [38]. Li *et al.* consider that the original traffic flow series contains linear and nonlinear parts, and predicted the linear part using ARIMA and the nonlinear part using RBF-NN. The results show that the hybrid model has better prediction results than the single ARIMA, RBF-NN model [39]. Compared with the single prediction model, the prediction accuracy of the hybrid model has been significantly improved. However, the existing studies still have shortcomings in the design, always ignoring the fact that traffic flow time series have typical periodic characteristics and will show different characteristics at different time granularity. This study begins to consider decomposing the original data series into sub-series with different frequencies or modes, building sub-prediction models, and combining the results of the sub-prediction models to obtain the final prediction results. To further demonstrate the model effect, two combined models of CNN and LSTM are compared to verify the prediction effect of this model.

III. METHODS

This section first introduces the CEEMDAN, SampEn, and LSTM. Based on this, a method that combines the three models was proposed.

A. CEEMDAN

The steps of using the CEEMDAN algorithm to decompose the travel time series are as follows.

Step 1: Add a series of adaptive white noise to the resampling travel time series:

$$T^i(t) = T(t) + \varphi_0 \theta^i(t), \quad i \in \{1, \dots, I\} \quad (1)$$

where $T^i(t)$ is the travel time series after adding white noise, $T(t)$ is the historical travel time series of road sections, $\theta^i(t)$ is the white noise added for the i th time, φ_0 is the noise coefficient, and I is the integration time (usually 10-20).

Step 2: Combined with EMD algorithm, the travel time series $T^i(t)$ is decomposed, and mean the decomposed IMF components:

$$d_1(t) = \frac{1}{I} \sum_{i=1}^I d_1^i \quad (2)$$

where, d_1^i is the i -th IMF component, and the $d_1(t)$ represents the first travel time series component.

Step 3: The first margin sequence can be obtained by removing $d_1(t)$ from $T(t)$:

$$r_1(t) = T(t) - d_1(t) \quad (3)$$

where, the $r_1(t)$ represents the first margin sequence.

Step 4: Continue to decompose $r_1(t) + \varphi_0 E_1[\theta^i(t)]$ and obtain the second travel time series component:

$$d_2(t) = \frac{1}{I} \sum_{i=1}^I E_1 \left\{ r_1(t) + \varphi_0 E_1[\theta^i(t)] \right\} \quad (4)$$

where, $E_j(\cdot)$ is the j -th IMF component obtained by EMD decomposition, and the $d_2(t)$ represents the second travel time series component.

Step 5: Calculate the remaining IMF components:

$$r_k(t) = r_{k-1}(t) - d_k(t), \quad k = 2, 3, \dots, K \quad (5)$$

$$d_{k+1}(t) = \frac{1}{I} \sum_{i=1}^I E_1 \left\{ r_k(t) + \varphi_k E_k[\theta^i(t)] \right\} \quad (6)$$

where, K is the total number of modes, $r_k(t)$ represents the k -th margin sequence, $d_{k+1}(t)$ represents the $(k+1)$ -th travel time series component, φ_k is the noise coefficient.

Step 6: When the travel time margin cannot be further decomposed. The margin was calculated as follows:

$$R(t) = T(t) - \sum_{k=1}^K d_k(t) \quad (7)$$

where, $R(t)$ represents the margin sequence.

B. SAMPLE ENTROPY (SampEn)

SampEn can measure the complexity of a time series well. Its calculation does not depend on the length of the travel-time subsequence and has excellent consistency. The smaller the sample entropy, the higher is the self-similarity of the time series; otherwise, the greater is the nonlinearity of the travel time series. Assuming that the travel time sequence $\{x(n)\} = x(1), x(2), \dots, x(N)$ consists of N sample time points, the calculation steps of the entropy are as follows.

Step 1: Convert the travel time series into a vector series with dimension m , $X_m(1), \dots, X_m(N - m + 1)$, and $X_m(i) = \{x(i), x(i+1), \dots, x(i+m-1)\}$, $1 \leq i \leq N - m + 1$. These vectors represent m consecutive values from the i th time-point.

Step 2: $d[X_m(i), X_m(j)]$ is the absolute value of the maximum difference in the corresponding element $X_m(i)$ and $X_m(j)$, which is calculated as follows:

$$d[X_m(i), X_m(j)] = \max_{k=0, \dots, m-1} (|x(i+k) - x(j+k)|) \quad (8)$$

Step 3: For a given $X_m(i)$, count the number of distances between $X_m(i)$ and $X_m(j)$ that is less than or equal to r , and record it as B_i . For $1 \leq i \leq N-m$, define:

$$B_i^m(r) = \frac{1}{N - m - 1} B_i \quad (9)$$

Step 4: Defined $B^{(m)}(r)$ as:

$$B^{(m)}(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} B_i^{(m)}(r) \quad (10)$$

Step 5: Increase the dimension to $m+1$, calculate the number with distance between $X_{m+1}(i)$ and $X_{m+1}(i)$ less than or equal to r , and record it as A_i . Defined $A_i^{(m)}(r)$ as:

$$A_i^{(m)}(r) = \frac{1}{N-m+1} A_i \quad (11)$$

Step 6: Defined $A_i^{(m)}(r)$ as:

$$A^{(m)}(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} A_i^{(m)}(r) \quad (12)$$

$B^{(m)}(r)$ is the probability that two sequences match m points under the similarity tolerance r . $A^{(m)}(r)$ is the probability that two sequences match $m+1$ points. The sample entropy is defined as

$$\text{SampEn}(m, r) = \lim_{N \rightarrow \infty} \left\{ -\ln \left[\frac{A^{(m)}(r)}{B^{(m)}(r)} \right] \right\} \quad (13)$$

C. LONG SHORT-TERM MEMORY (LSTM)

In the field of traffic prediction, the traditional recurrent neural network has a poor ability to predict time series with a long-time delay, and the problem of long-term dependence will appear. LSTM is a type of temporal recurrent neural network that is specially designed to solve the long-term dependency problem of general RNN. It is typically used to predict temporal series data. The neural network module of LSTM has four layers that interact with information in a special manner; its structure is shown in Figure 1. Among them, the forgetting gate is responsible for controlling the choice of information in long-term memory. The closer to 1 means the more information is retained; The input gate is used to control the input process of the current time information, that is, to update the memory cell; The output gate controls the inflow of timing information in memory cells and determines the short-term memory part. There are two states in LSTM, one is long-term memory state C and the other is short-term memory state H . C will update itself through the information input at each time sequence and deliver it all the time; H extracts the corresponding memory at each time step.

The calculation of each LSTM layer can be explained using the following formula:

$$i^{(t)} = \sigma(W^{(i)}x^{(t)} + U^{(i)}h^{(t-1)}) \quad (14)$$

$$f^{(t)} = \sigma(W^{(f)}x^{(t)} + U^{(f)}h^{(t-1)}) \quad (15)$$

$$o^{(t)} = \sigma(W^{(o)}x^{(t)} + U^{(o)}h^{(t-1)}) \quad (16)$$

$$\bar{c}^{(t)} = \tanh(W^{(c)}x^{(t)} + U^{(c)}h^{(t-1)}) \quad (17)$$

$$c^{(t)} = f^{(t)} \times c^{(t-1)} + i^{(t)} \times \bar{c}^{(t)} \quad (18)$$

$$h^{(t)} = o^{(t)} \times \tanh(c^{(t)}) \quad (19)$$

where, $x^{(t)}$ is the input signal, $\sigma(\cdot)$ is the activation function, $i^{(t)}$, $f^{(t)}$, $o^{(t)}$ are Input gate, Forget gate and Output

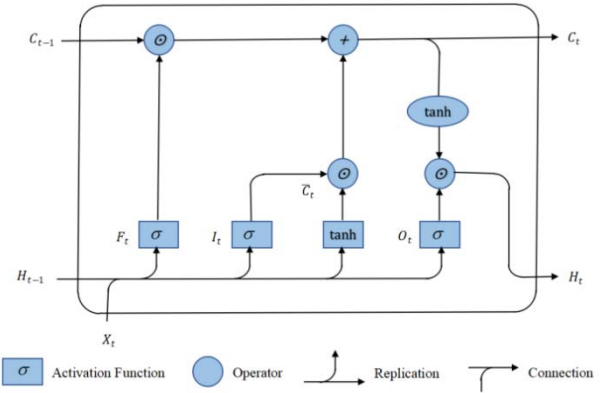


FIGURE 1. Network structure of LSTM.

gate, $\bar{c}^{(t)}$, $c^{(t)}$ are New memory cell and Final memory cell, $h^{(t)}$ is Hidden state, $W^{(i)}$, $U^{(i)}$, $W^{(f)}$, $U^{(f)}$, $W^{(o)}$, $U^{(o)}$, $W^{(c)}$, $U^{(c)}$ are parameter matrix.

D. COMBINED FORECASTING MODEL FRAMEWORK

Based on the RNN, combined with the CEEMDAN and SampEn algorithms, the framework of the combined prediction model was constructed, as shown in Figure 2. Firstly, the expressway gantry data obtained by the fixed collection technology is cleaned, and the abnormal value is removed. According to the characteristics of the change of the traffic state of the expressway, a reasonable sampling time window is set, and the “ 3σ ” criterion method, box chart method and 2SD-CE algorithm respectively remove outliers from the travel time series for 7 consecutive days, and the Min Max Scaler in Scikit-learn is used to normalize the data within the range of $[-1, 1]$. After a series of preprocessing, a relatively accurate historical travel time series $T(t)$ is obtained, which is the premise of successful prediction.

Then, CEEMDAN algorithm is applied to decompose the travel time series of road sections, and decompose them into several IMFs at different time scales. The final travel time series can be expressed as:

$$T(t) = \sum_{k=1}^K d_k(t) + R(t) \quad (20)$$

where $R(t)$ is the margin sequence decomposed by CEEMDAN algorithm.

Next, the complexity of time-series components of road sections is calculated by SampEn algorithm. Through the sample entropy in part A of this section, the sample entropy can be estimated by the following equation:

$$\text{SampEn}(m, r, N) = -\ln \left[\frac{A^m(r)}{B^m(r)} \right] \quad (21)$$

where N is the number of data points in the travel time series.

The threshold value of sample entropy is set to divide the frequency of each sub sequence. According to the

randomness from high to low, it is divided into high-frequency sequence, intermediate frequency sequence and low-frequency sequence.

In the prediction of each sub sequence, the high-frequency sequence adopts bidirectional LSTM (BiLSTM), the intermediate frequency sequence adopts LSTM, and the low-frequency sequence adopts vanilla RNN. Finally, the prediction results are added to obtain the travel time prediction value. This combined prediction model is referred to as combination model 1 (CM1). As a comparison, CEEMDAN and EMD algorithms are respectively used to decompose the time series without frequency division, and then LSTM algorithm is used to predict these sub sequences. Finally, the final prediction results are obtained by summing all the prediction results. The models are respectively CM2 and CM3. Later, this will be demonstrated by a case study.

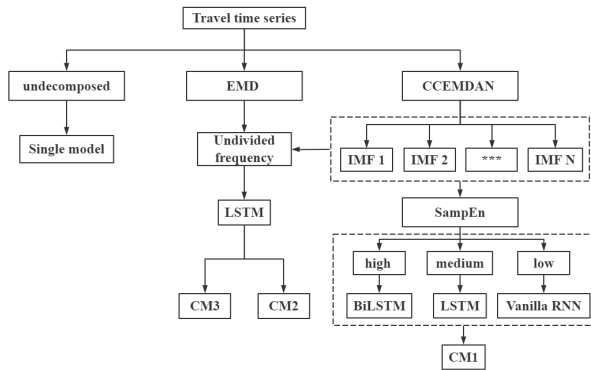


FIGURE 2. Combined forecasting model framework.

IV. CASE STUDY

A. DATA SOURCE

The travel time data of the expressway section is collected by using the fixed acquisition technology to read the vehicle information in the vehicle electronic tag through the ETC (Electronic Toll Collection) microwave antenna installed on the gantry of the expressway toll station. The data range includes expressways in Chengdu and Ya’an City, Sichuan Province, including the Yalu, Chengnan, and Chengya expressways. The specific locations are shown in Figure 3. The time range is from July 30 to September 30, 2020, with a total of 1.2 billion pieces of data. The field names of the original data include CENTERNAME, GANTRYNAME, PILENUMBER, GANTRYORDERNUM, TRANSTIME, VLP, and VEHICLETYPE. See Table 1 for the meanings of these fields. Among all vehicle types, the number of class 1 passenger cars (passenger cars with fewer than seven seats) accounts for 85%, and their operation condition can represent the traffic condition of the road section. To eliminate the influence of the dynamic performance of different types of vehicles on travel time, only the travel time of class 1 passenger cars was used and predicted. The original data for the expressway gantry are listed in Table 2.

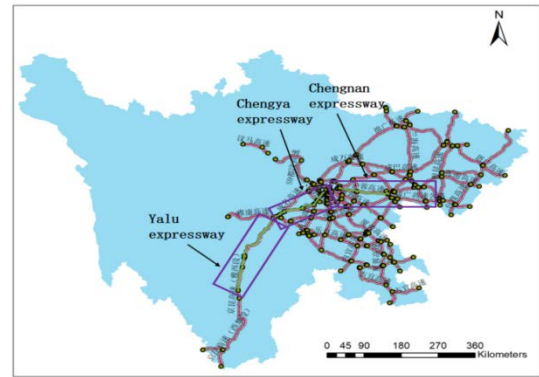


FIGURE 3. Sichuan expressway network.

TABLE 1. Meaning of each field of original data.

| Initial field | Data type | Field meaning |
|----------------|-----------|--------------------------------------|
| CENTERNAME | Char | Expressway name |
| GANTRYNAME | Char | Gantry name |
| PILENUMBER | Char | Stake number |
| GANTRYORDERNUM | Int | Up and down directions of Expressway |
| TRANSTIME | Datetime | Time of vehicle passing |
| VLP | Char | Car number |
| VEHICLETYPE | Int | Vehicle type |

TABLE 2. Original data (Excerpt).

| CENTERNAME | GANTRYNAME | PILENUMBER | GANTRYORDERNUM | TRANSTIME | VLP | VEHICLETYPE |
|------------|------------------------|------------|----------------|-----------------|---------|-------------|
| Yalu | Yihai-,enghuo Gantry | k2132+015 | 201 | 2020/9/29 11:18 | LDXX XX | 1 |
| Yalu | Menghuo-Shimian Gantry | k2095+050 | 201 | 2020/9/29 11:55 | LDXX XX | 1 |
| Yalu | Shimian-Haiyuan Gantry | k2063+960 | 201 | 2020/9/29 12:18 | LDXX XX | 1 |
| Yalu | Hainan-Haibei Gantry | K2038+850 | 201 | 2020/9/29 12:36 | LDXX XX | 1 |
| Yalu | Hainan-Haibei Gantry | K2035+960 | 201 | 2020/9/29 12:38 | LDXX XX | 1 |
| Yalu | Haibei-Hainan Gantry | K2035+900 | 101 | 2020/9/30 14:43 | LDXX XX | 1 |

B. DATA CLEANING

In the process of gantry data acquisition and transmission, owing to problems with the system itself or other factors, data errors can easily occur. In the process of extracting travel time from gantry data, firstly through simple screening rules to eliminate obviously wrong data, such as data record repeat, travel time is negative. The following rules are mainly included.

Rule 1, excludes data with a model data of 0.

Rule 2, removes data at the exit or entrance of a certain section.

Rule 3, removes data with an empty gantry name.

Rule 4, removes data with a vehicle entry time later than or equal to the exit time. Such data is not of research significance, so it is excluded.

Rule 5, removes data whose travel time is significantly beyond the range (greater than one day). The long travel time may be caused by the driver stopping in the service area while driving at high speed.

Rule 6, removes data with a stroke time interval of less than 10 seconds. A short travel time may be because the vehicle is recorded twice by the same gantry, or simultaneously by the opposite gantry.

Rule 7, removes duplication of data records.

Rule 8, exclude cases where the travel time is negative or empty.

After cleaning the obviously incorrect data, there are still some outliers in the travel time series. Travel time series have the characteristics of high randomness and nonlinearity. It is unreasonable to combine the travel times at all the time points to eliminate outliers. It is necessary to set a reasonable sampling time window according to different traffic conditions. In general, a larger time window cannot effectively capture the changes in traffic state within a smaller time range to eliminate the normal value of a larger travel time. Although a smaller time window can reasonably obtain the characteristics of the travel time change, it is vulnerable to the influence of the sample size. Therefore, according to the characteristics of the change in expressway traffic state, 10 min was selected as the time window in the evening peak period (16:00 to 22:00) to capture the time-varying difference of travel time, 30 min was selected as the time window of early morning and night time (0:00 to 6:00), and the time window of other peak periods was set to 20 min.

The 2SD-CE algorithm was used to filter abnormal travel time. The principle is that when the difference between the sample value of the travel time and the average value of the total sample is greater than twice the standard deviation of the total sample, the sample value of the travel time is classified as an abnormal value and is eliminated. The specific screening requirements are shown in Figure 4.

The 2SD-CE algorithm is used to filter the abnormal travel time with the specific steps as follows.

Step 1: According to the travel time characteristics of the road section and different traffic conditions, reasonable statistical time windows were set.

Step 2: Calculate the mean μ and standard deviation σ of the travel time samples in each statistical time window.

Step 3: Judge whether there are any data outside the range of $\mu \pm 2\sigma$ in the sample. If there is, remove the data and refer to Step2 for recalculation. If it does not exist, proceed to Step4;

Step 4: Retain the travel time sample after final screening.

C. DATA CLEANING RESULTS

All the sections of the Yalu Expressway were selected for outlier elimination. Taking section 1 in the upward direction

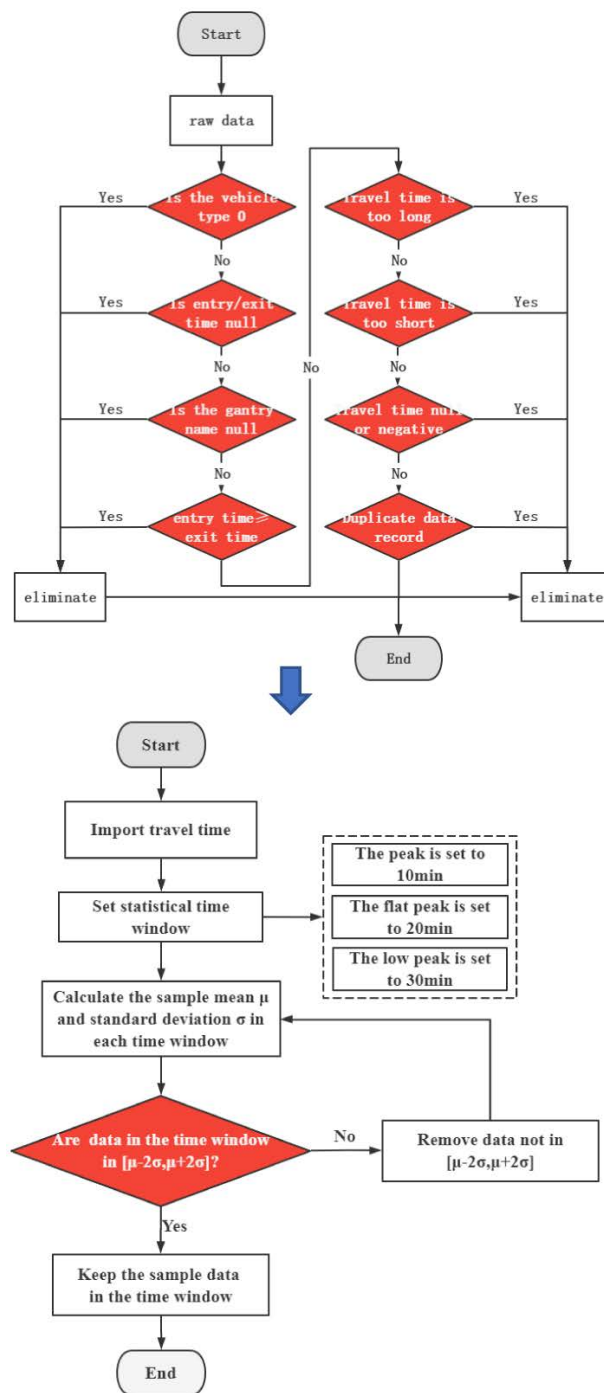


FIGURE 4. Data cleaning process.

of Yalu expressway as an example, according to the above travel time data cleaning method, the scatter diagram of travel time distribution after eliminating obvious wrong data can be obtained, as shown in Figure 5 (b). It can be seen from the figure that although the travel time shows a certain trend after the initial cleaning of the data with an obvious wrong travel time, there are still many noise points; that is, there are still some data with large travel times in the flat peak. If an appropriate outlier elimination method is not adopted, it is

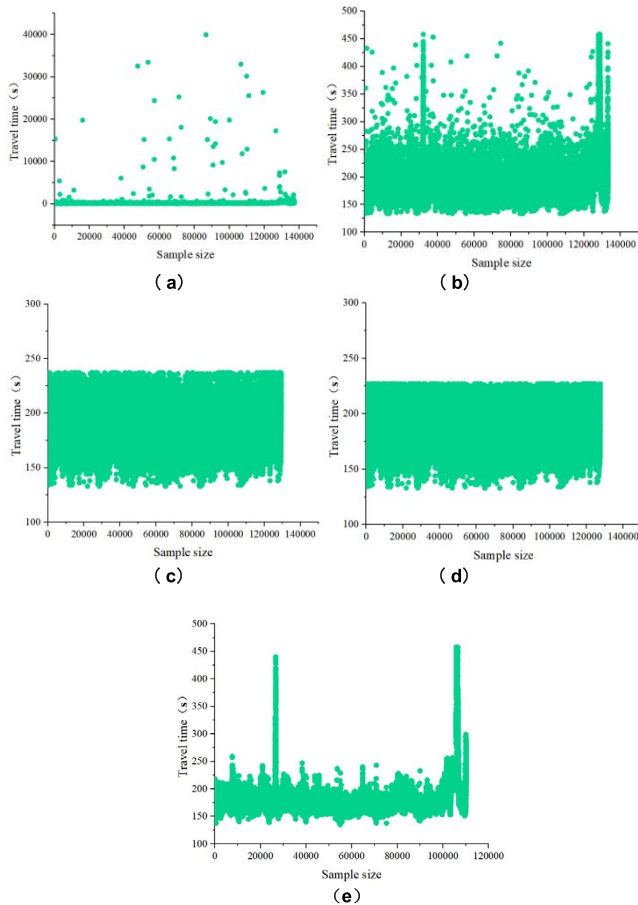


FIGURE 5. Data cleaning results,(a)raw travel time data(b)after eliminating obvious wrong data(c)“3σ”Criterion method(d)box diagram method(e)2SD-CE algorithm.

bound to affect the quality of the training samples and reduce the effect of model fitting.

Use “3σ” criterion method, box chart method and 2SD-CE algorithm eliminate outliers from the travel time series for 7 consecutive days respectively. The elimination effect is shown in Figure 5 (c), (d) and (e). As can be seen from the figure, although the “3σ” criterion method and box chart method can effectively eliminate the noise points with excessive travel time in the peak period, they also eliminate the characteristic points with excessive travel time in the peak period, and do not retain the characteristics of excessive travel time in the peak period. The effect of eliminating the abnormal data of travel time in the expressway section is not ideal, and there are some limitations; therefore, they are not suitable for the treatment of the abnormal value of travel time in the expressway section. The 2SD-CE algorithm subdivides each time period by setting a statistical time window to retain the travel time characteristics in each time period. It can be seen from Figure 5 (e) that after cleaning the data using this method, not only are the outliers of travel time effectively removed, but the trend of excessive travel time in peak hours is also retained to a great extent. Therefore, the 2SD-CE algorithm was applied to address the outliers of travel time.

The cleaned data were resampled to a time granularity of 5 min, and there were few missing data points. The historical average method was used to fill in missing values. Because the travel time series generally fluctuates significantly, the Min Max Scaler in Scikit-learn is used to normalize the data within the range of $[-1, 1]$.

V. PERFORMANCE EVALUATION

A. MODEL EVALUATION INDEX

Historical travel time series based on expressway gantry data can predict future travel time after preprocessing. However, to ensure that the prediction results meet the needs of expressway intelligent control and guidance, it is often necessary to measure whether the model meets certain requirements, such as accuracy, efficiency, and portability. Therefore, to measure the performance of different models, the root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) were used as evaluation indices. The smaller the values of RMSE, MAE, and MAPE, the better is the prediction performance of the model. This is specified as the observed value y_i and the predicted value \hat{y}_i , $i \in (1, \dots, n)$. The evaluation indices are defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (22)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (23)$$

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (24)$$

B. PARAMETER SETTING

The experiment was implemented in Python IDE PyCharm, and the main software environments used were Keras Version 2.4.3 and TensorFlow Version 2.3.0. The hardware environment was a Lenovo Y7000 personal computer (Intel Core i7-9750H CPU@2.60GHz processor, NVIDIA GeForce GTX 1660ti graphics card, 16GB RAM, 512 GB hard disk). When setting the model training parameters, we set the ratio of the training and test sets of the total samples to 9:1.

The grid search method of internal nested cross-validation was used to optimize the network parameters. The steps of this method are as follows: first, the pre-selected values of various parameters such as the number of neurons in the hidden layer, learning rate and batch size are listed in the dictionary, and the “grid” is generated by the exhaustive method through the arrangement and combination of various parameters, and then put into the model in batches for prediction performance evaluation, In each batch, the cross validation method is used to fully evaluate the model performance of a single group of parameters, and finally the optimal parameter combination is selected through comparison.

Through the grid search method, in the vanilla RNN model, the number of neurons is determined to be 32, and the learning

rate is 0.001, set batch_size to 64 and the number of iterations to 20; In the LSTM model, the number of neurons is determined to be 48, and the learning rate is 0.001, set batch_size to 64 and the number of iterations to 30; In the BiLSTM model, the number of neurons is 48 and the learning rate is 0.001, batch_size is 128 and the number of iterations is set to 30. The input length of the historical time series of all the models was 15. The mean square error (MSE) was used as the loss function, and the adaptive moment estimation optimization gradient descent algorithm (Adam) was used as the optimizer.

C. TRAVEL TIME SERIES DECOMPOSITION RESULTS

The CEEMDAN algorithm was used to decompose the travel time series of all sections of the Yalu expressway into several IMFs at different time scales. Taking the travel time data of section 2 with 5 min as the time granularity for seven consecutive days as an example, the decomposition results are shown in Figure 6, from top to bottom are the original time series and the 9 IMFs components sequentially decomposed by the CEEMDAN algorithm. It can be seen from the figure that the decomposed travel time series eliminates the noise of the original sequence to a great extent. In addition, the decomposed travel time series has a certain smoothness, which can significantly improve the prediction effect of the travel series.

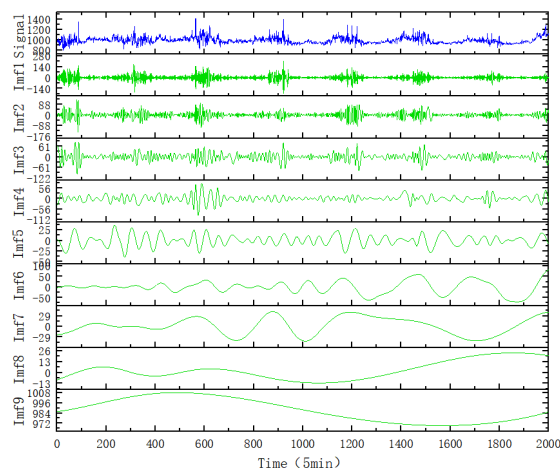


FIGURE 6. Travel time series decomposition results.

After decomposing the travel time series into several IMFs with the CEEMDAN algorithm, the complexity of the time series components of all sections of the Yalu expressway is calculated using SampEn. The mean values of the calculation results are presented in Table 3. It can be seen that the sample entropy gradually decreases from IMF1 to IMF9, which means that the complexity of the travel time series also decreases. Set the threshold value of sample entropy as 1 and 0.2, then IMF1 ~ IMF4 are divided into high-frequency sequences with high randomness, IMF5–IMF7 are intermediate frequency sequences with general randomness, and IMF8

and IMF9 are divided into low-frequency sequences with relatively low randomness.

TABLE 3. Sample entropy of travel time subsequences after CEEMDAN decomposition.

| IMF | IMF1 | IMF2 | IMF3 | IMF4 |
|----------------|---------|---------|---------|---------|
| Sample entropy | 4.18965 | 4.07753 | 2.34770 | 1.11358 |
| IMF5 | IMF6 | IMF7 | IMF8 | IMF9 |
| 0.72465 | 0.43551 | 0.34570 | 0.12120 | 0.07081 |

To evaluate the prediction performance and stability of CM1, two combined models of convolutional neural network (CNN) and LSTM were selected: CNN-LSTM and Convolutional LSTM (ConvLSTM). The specific parameters of the various control models are set as follows:

- (1) LSTM: The number of neurons was set to 48, the learning rate was 0.001, the batch_size was set to 64, and the number of iterations was 30.
- (2) CNN-LSTM: Uses the convolution part of the model to process the data and inputs the processed one-dimensional array into the LSTM model. In the parameter setting, the number of filters was 64, and the size of the convolution kernel was 1×2 . The activation function is a linear rectified linear unit (ReLU).
- (3) ConvLSTM: The multiplication operation in each gate of the LSTM unit is replaced by a convolution operation. The difference from the CNN-LSTM model is that the former does not convolute the cycle time kernel. In the parameter setting, the number of filters was 64, and the size of the convolution kernel was 1×2 . The activation function was ReLU.

D. PREDICTION EFFECT UNDER DIFFERENT TIME GRANULARITY

To analyze the short-term prediction effect of the combined model on the travel time of expressways, we selected the time granularity of 5 min, 10 min, and 15 min to predict the travel time of all sections of the Yalu expressway. For example, in Section 2, the prediction results are shown in Figure 7. It can be observed from the figure that the three combined models show a good prediction effect of travel time under different granularities. Compared with CM1 and CM2 using the CEEMDAN algorithm, CM3 using the EMD algorithm overestimates the travel time value at most time points, especially in the 5-minute time granularity. Both the CM1 and CM2 models can better conform to the changing trend of the travel time series and have high prediction accuracy.

For a more accurate comparison of the differences between the models, the MAPE box diagram of the prediction results of each model under the time granularity of 5, 10, and 15 min is obtained, as shown in Figure 8. According to the figure, under the time granularity of 5, 10, and 15 min, the MAPE of CM1, CM2, and CM3 at each time point was maintained within 7%, 8%, and 15%, respectively, indicating that these three models have high prediction accuracy at each time

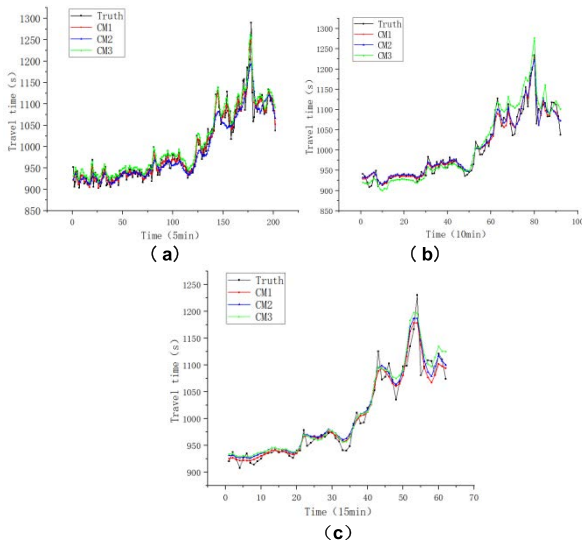


FIGURE 7. Prediction results of different models with different time granularities: (a) time granularity: 5 min; (b) time granularity: 10 min; (c) time granularity: 15 min.

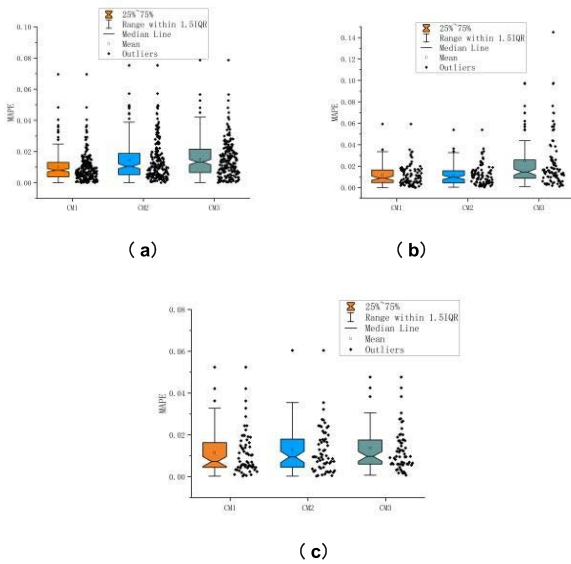


FIGURE 8. MAPE error diagram of different models with different time granularity, (a) time granularity is 5 minutes (b) time granularity is 10 minutes (c) time granularity is 15 minutes.

point. Under different time granularities, the median and mean of MAPE of CM1 and CM2 are lower than that of CM3, and there are few data points with MAPE close to 0 in CM3; therefore, its prediction error is also relatively large.

To further reflect the prediction effect of CM1, two combined models of CNN and LSTM were introduced: CNN-LSTM and ConvLSTM. The errors of the prediction results of different models under the time granularity of 5, 10, and 15 min are shown in Table 4. It can be seen that under different time granularities, the prediction effect of ConvLSTM is better than that of CNN-LSTM, and the prediction performance of LSTM is not inferior to that of the combined model of CNN and LSTM, which shows that under the condition

of a single feature, the combination of CNN and LSTM cannot significantly improve the prediction accuracy of RNN.

TABLE 4. Prediction error of different models with different granularity.

| | Prediction model | CNN-LSTM | Conv-LSTM | LSTM | CM3 | CM2 | CM1 |
|------------------------------|------------------|----------|-----------|--------|--------|--------|--------|
| 5-minute s time granularity | RMSE | 47.173 | 32.454 | 31.844 | 19.533 | 17.699 | 14.376 |
| | MAE | 33.181 | 25.736 | 22.357 | 15.132 | 13.068 | 10.152 |
| | MAPE (%) | 3.175 | 2.567 | 2.151 | 1.513 | 1.366 | 0.997 |
| 10-minute s time granularity | RMSE | 51.987 | 39.363 | 42.004 | 26.374 | 15.677 | 15.554 |
| | MAE | 34.634 | 26.219 | 28.062 | 17.929 | 11.692 | 11.638 |
| | MAPE (%) | 3.232 | 2.456 | 2.618 | 1.679 | 1.145 | 1.137 |
| 15-minute s time granularity | RMSE | 45.071 | 32.518 | 31.002 | 19.386 | 17.528 | 16.807 |
| | MAE | 31.581 | 22.124 | 22.233 | 14.224 | 12.991 | 12.004 |
| | MAPE (%) | 3.019 | 2.117 | 2.153 | 1.377 | 1.261 | 1.152 |

Under the time granularity of 5 min, 10 min, and 15 min, the RMSE of CM1 using the decomposition algorithm for the travel time series was 54.9%, 62.9%, and 45.8% lower than that of LSTM, the MAE was 54.5%, 58.5%, and 46% lower, and the MAPE was 53.7%, 56.6%, and 46.5% lower, respectively. CM1 model is better than CM2 and CM3, which can reduce the RMSE by 18.8 ~ 26.4%, 0.8 ~ 41%, 4.1 ~ 13.3%, MAE by 22.3 ~ 32.9%, 0.5 ~ 35.1%, 7.6 ~ 15.6%, MAPE by 27% ~ 34.1%, 0.7% ~ 32.3%, 8.68% ~ 16.3%.

Therefore, compared with the combined prediction mode of CNN and LSTM, the combined prediction mode of step-by-step prediction using the time-series decomposition algorithm has a higher prediction accuracy. In addition, the advantage of CM1 over CM2 is that it uses the BiLSTM model to predict high-frequency travel time subsequences. Compared with the LSTM model, the BiLSTM model can better predict high-frequency travel time subsequences.

E. PREDICTION EFFECT UNDER DIFFERENT EXPRESSWAY

To better analyze the applicability of the model to different expressway datasets, the travel time on September 30 was selected as the prediction object, the time granularity was set to 5 min, and the traffic volume of each section of the Yalu expressway, Chengnan Expressway, and Chengya expressway were counted. According to the statistical results, sections with large and small traffic flows on the Yalu expressway were determined. The CM1 model was used to

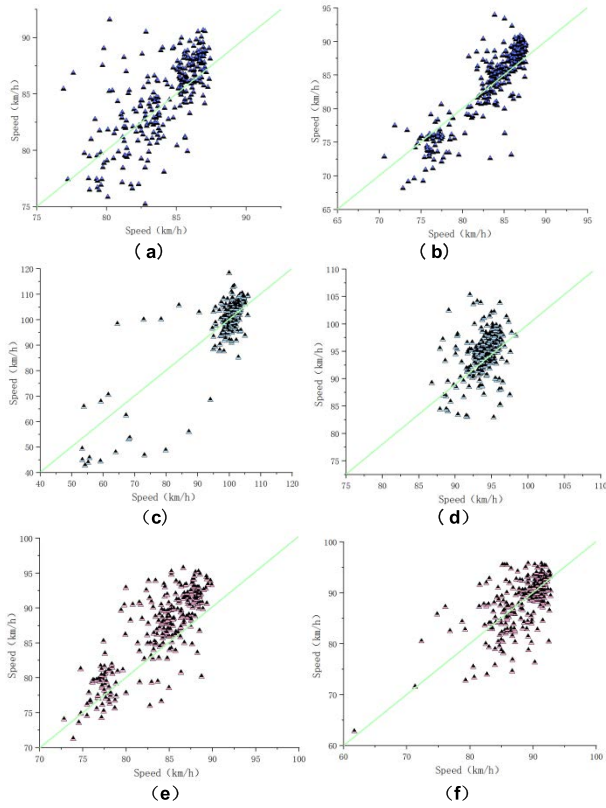


FIGURE 9. Scatter diagram of prediction results of different expressway sections, (a)large traffic flow of Yalu expressway(b)small traffic flow of Yalu expressway (c) large traffic flow of Chengnan expressway (d)small traffic flow of Chengnan expressway(e)large traffic flow of Chengya expressway(f)small traffic flow of Chengya expressway.

predict the travel time. The measured speed value and the speed value converted from the predicted value of travel time were drawn into a scatter diagram to analyze the prediction effect of the model. The results are shown in Figure 9. Each figure contained 288 data points. The straight line represents the 45 °line, the X-coordinate is the predicted value, and the Y-coordinate is the real value. The closer it is to the 45 °line, the more accurate the prediction result.

Based on the prediction results of the whole day on September 30, it can be seen that the data points of the sections with small traffic flow on the Yalu expressway, Chengnan expressway, and Chengya expressway are closer to the 45° line, and the speeds are basically distributed between 70 ~ 100 km/h, so the prediction results are ideal, the distribution of prediction data points in the sections with large traffic flow is relatively discrete, and the prediction effect is worse than that in the sections with small traffic flow.

The prediction error results of the different prediction models for the Yalu, Chengnan, and Chengya expressways are shown in Figure 10. Among them, the section with a large traffic flow is represented by M, and the section with a small traffic flow is represented by L. Overall, the prediction effect of CM1 and CM2 in each expressway section is better than that of the other models. The CM1 model can more accurately grasp the periodicity and regularity of travel times in different

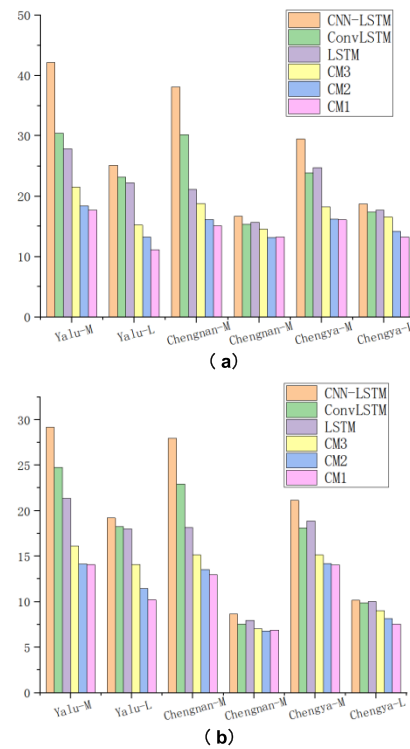


FIGURE 10. Error diagram of prediction results of different expressway sections, (a) RMSE (b) MAE.

sections of traffic flow. The prediction effect of each model in the section with a small traffic flow was significantly better than that in the section with a large traffic flow. This is because the change in vehicle speed was more frequent in the section with a large traffic flow. Sometimes, it is difficult to capture the operation situation of slow driving during peak hours, resulting in differences in travel time prediction results.

VI. CONCLUSION AND FUTURE WORK

Expressway travel time data series have significant nonlinear and nonstationary characteristics, and it is difficult for a single prediction model to meet the increasing demand for prediction accuracy. This study attempts to combine the CEEMDAN algorithm with an RNN to build a prediction model, which is verified by different time granularities and expressways. The main conclusions are as follows. (1) The CM1 combined prediction model proposed in this paper has high accuracy for travel time prediction of different time granularities and expressways, and the prediction model has certain generalization and robustness. (2) Under the time granularity of 5 min, 10 min, and 15 min, the prediction performances of CM1, CM2, and CM3 were better than those of LSTM, CNN-LSTM, and ConvLSTM, which indicate that the combined prediction method was better than the single model. (3) Under different time granularity, CM1 is better than CM2 and CM3; under different expressway, CM1 can more accurately grasp the variation characteristics of travel time in different sections of traffic flow. This indicates that the

BiLSTM model can better predict the high-frequency travel time subsequence.

However, this paper still has some limitations. Firstly, the CM1 combined model training time is long because the introduction of CEEMDAN algorithm affects the training time of LSTM model. Secondly, the model only considers the randomness and periodicity of traffic volume sequence, but ignores other factors that affect traffic volume under real conditions, such as weather and traffic changes in adjacent areas. To improve the stroke time prediction model, the future work direction will focus on the following two aspects: (1) perform a more comprehensive time complexity analysis using asymptotic notations to precisely show the running time of the work, find effective ways to improve computing efficiency, (2) more comparative analysis of the predictive effect of CEEMDAN and deep neural structures should be conducted, explore and improve the algorithm to include more influencing factors, and extend to the road network level.

REFERENCES

- [1] Y. Tu, S. Lin, J. Qiao, and B. Liu, "Deep traffic congestion prediction model based on road segment grouping," *Appl. Intell.*, vol. 51, pp. 8519–8541, Nov. 2021, doi: [10.1007/s10489-020-02152-x](https://doi.org/10.1007/s10489-020-02152-x).
- [2] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, 2017, doi: [10.1049/iet-its.2016.0208](https://doi.org/10.1049/iet-its.2016.0208).
- [3] J. Zheng and M. Huang, "Traffic flow forecast through time series analysis based on deep learning," *IEEE Access*, vol. 8, pp. 82562–82570, 2020, doi: [10.1109/ACCESS.2020.2990738](https://doi.org/10.1109/ACCESS.2020.2990738).
- [4] B. Wang, J. Wang, and Z. Zhang, "Traffic flow prediction model based on deep learning," in *Proc. 21st Int. Conf. Man-Mach.-Environ. Syst. (MMESE)*, in Lecture Notes in Electrical Engineering, vol. 800, S. Long and B. S. Dhillon, Eds. Singapore: Springer, 2022, pp. 739–745, doi: [10.1007/978-981-16-5963-8_100](https://doi.org/10.1007/978-981-16-5963-8_100).
- [5] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.
- [6] W. H. Luo, B. T. Dong, and Z. S. Wang, "Short-term traffic flow prediction based on CNN-SVR hybrid deep learning model," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 17, no. 5, pp. 68–74, Oct. 2017, doi: [10.16097/j.cnki.1009-6744.2017.05.010](https://doi.org/10.16097/j.cnki.1009-6744.2017.05.010).
- [7] A. Boukerche and J. Wang, "Machine learning-based traffic prediction models for intelligent transportation systems," *Comput. Netw.*, vol. 181, no. 3, Nov. 2020, Art. no. 107530, doi: [10.1016/j.comnet.2020.107530](https://doi.org/10.1016/j.comnet.2020.107530).
- [8] H. Huang, J. Chen, X. Huo, Y. Qiao, and L. Ma, "Effect of multi-scale decomposition on performance of neural networks in short-term traffic flow prediction," *IEEE Access*, vol. 9, pp. 50994–51004, 2021, doi: [10.1109/ACCESS.2021.3068652](https://doi.org/10.1109/ACCESS.2021.3068652).
- [9] W. Li, S. Chen, X. Wang, C. Ying, and Z. Huang, "A hybrid approach for short-term traffic flow forecasting based on similarity identification," *Mod. Phys. Lett. B*, vol. 35, no. 13, Feb. 2021, Art. no. 2150212, doi: [10.1142/S0217984921502122](https://doi.org/10.1142/S0217984921502122).
- [10] I. Chabini, "Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1645, no. 1, pp. 170–175, Jan. 1998, doi: [10.3141/1645-21](https://doi.org/10.3141/1645-21).
- [11] L. Cai, Y. Yu, S. Zhang, Y. Song, Z. Xiong, and T. Zhou, "A sample-rebalanced outlier-rejected k -nearest neighbor regression model for short-term traffic flow forecasting," *IEEE Access*, vol. 8, pp. 22686–22696, 2020, doi: [10.1109/ACCESS.2020.2970250](https://doi.org/10.1109/ACCESS.2020.2970250).
- [12] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. B, Methodol.*, vol. 18, no. 1, pp. 1–11, Feb. 1984.
- [13] X. Liu, "Application of ARMA time series model," *Techn. Autom. Appl.*, vol. 8, pp. 65–66, Aug. 2008.
- [14] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box–Jenkins techniques," *Transp. Res. Rec.*, vol. 773, no. 722, pp. 1–9, Jan. 1979.
- [15] W. Li, J. Li, and T. Wang, "Improved ARIMA model traffic flow prediction method based on box-cox exponential transformation," *J. Wuhan Univ. Technol.*, vol. 44, no. 6, pp. 974–977, 2020, doi: [10.3963/j.issn.2095-3844.2020.06.006](https://doi.org/10.3963/j.issn.2095-3844.2020.06.006).
- [16] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved k -nearest neighbor model for short-term traffic flow prediction," *Proc., Social Behav. Sci.*, vol. 96, pp. 653–662, Nov. 2013, doi: [10.1016/j.sbspro.2013.08.076](https://doi.org/10.1016/j.sbspro.2013.08.076).
- [17] B. An and H. Zhou, "Short-term bus arrival time prediction based on chaos theory," *J. Xiamen Univ.*, vol. 57, no. 4, pp. 565–571, 2018, doi: [10.6043/j.issn.0438-0479.201710001](https://doi.org/10.6043/j.issn.0438-0479.201710001).
- [18] J. E. Disbro and M. Frame, "Traffic flow theory and chaotic behavior," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1225, pp. 109–115, Jan. 1989.
- [19] J. Wang and Q. Shi, "The summary of the short-term traffic flow predicts model," *China Public Secur.*, vol. 1, pp. 92–98, May 2005.
- [20] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004, doi: [10.1109/TITS.2004.837813](https://doi.org/10.1109/TITS.2004.837813).
- [21] H. Su, L. Zhang, and S. Yu, "Short-term traffic flow prediction based on incremental support vector regression," in *Proc. 3rd Int. Conf. Natural Comput. (ICNC)*, 2007, pp. 640–645, doi: [10.1109/ICNC.2007.661](https://doi.org/10.1109/ICNC.2007.661).
- [22] C. Luo, C. Huang, J. Cao, J. Lu, W. Huang, J. Guo, and Y. Wei, "Short-term traffic flow prediction based on least square support vector machine with hybrid optimization algorithm," *Neural Process. Lett.*, vol. 50, pp. 2305–2322, Mar. 2019, doi: [10.1007/s11063-019-09994-8](https://doi.org/10.1007/s11063-019-09994-8).
- [23] P. B. Weerakody, K. W. Wong, G. Wang, and W. Ela, "A review of irregular time series data handling with gated recurrent neural networks," *Neurocomputing*, vol. 441, pp. 161–178, Jun. 2021, doi: [10.1016/j.neucom.2021.02.046](https://doi.org/10.1016/j.neucom.2021.02.046).
- [24] M. I. Sameen and B. Pradhan, "Severity prediction of traffic accidents with recurrent neural networks," *Appl. Sci.*, vol. 7, no. 6, pp. 476–486, 2017, doi: [10.3390/app7060476](https://doi.org/10.3390/app7060476).
- [25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [26] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015, doi: [10.1016/j.trc.2015.03.014](https://doi.org/10.1016/j.trc.2015.03.014).
- [27] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Math. Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998, doi: [10.1098/rspa.1998.0193](https://doi.org/10.1098/rspa.1998.0193).
- [28] W. Zhang, Z. Qu, K. Zhang, W. Mao, Y. Ma, and X. Fan, "A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting," *Energy Convers. Manag.*, vol. 136, pp. 439–451, Mar. 2017, doi: [10.1016/j.trc.2015.03.014](https://doi.org/10.1016/j.trc.2015.03.014).
- [29] M. Duo, Y. Qi, G. Lina, and E. Xu, "A short-term traffic flow prediction model based on EMD and GPSSO-SVM," in *Proc. IEEE 2nd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Mar. 2017, pp. 2554–2558, doi: [10.1109/IAEAC.2017.8054485](https://doi.org/10.1109/IAEAC.2017.8054485).
- [30] S. Du, Z. Xu, and J. Lv, "An EMD- and GRU-based hybrid network traffic prediction model with data reconstruction," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–7, doi: [10.1109/ICC-Workshops50388.2021.9473822](https://doi.org/10.1109/ICC-Workshops50388.2021.9473822).
- [31] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 1, no. 1, pp. 1–41, 2009, doi: [10.1142/S1793536909000047](https://doi.org/10.1142/S1793536909000047).
- [32] J. Tang, X. Chen, Z. Hu, F. Zong, C. Han, and L. Li, "Traffic flow prediction based on combination of support vector machine and data denoising schemes," *Phys. A, Stat. Mech. Appl.*, vol. 534, Nov. 2019, Art. no. 120642, doi: [10.1016/j.physa.2019.03.007](https://doi.org/10.1016/j.physa.2019.03.007).
- [33] J. Liu, N. Wu, Y. Qiao, and Z. Li, "Short-term traffic flow forecasting using ensemble approach based on deep belief networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 404–417, Jan. 2022, doi: [10.1109/TITS.2020.3011700](https://doi.org/10.1109/TITS.2020.3011700).
- [34] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 4144–4147, doi: [10.1109/ICASSP.2011.5947265](https://doi.org/10.1109/ICASSP.2011.5947265).

- [35] Y. Guo, X. Cao, B. Liu, and K. Peng, "Chaotic time series prediction using LSTM with CEEMDAN," *J. Phys., Conf. Ser.*, vol. 1617, Aug. 2020, Art. no. 012094, doi: [10.1088/1742-6596/1617/1/012094](https://doi.org/10.1088/1742-6596/1617/1/012094).
- [36] W. Lu, Y. Rui, Z. Yi, B. Ran, and Y. Gu, "A hybrid model for lane-level traffic flow forecasting based on complete ensemble empirical mode decomposition and extreme gradient boosting," *IEEE Access*, vol. 8, pp. 42042–42054, 2020, doi: [10.1109/ACCESS.2020.2977219](https://doi.org/10.1109/ACCESS.2020.2977219).
- [37] Y. Zhu, K. Zhang, and C. Zhao, "Short-term traffic flow prediction of expressway based on CEEMDAN-PE-BiGRU combined model optimized by attention mechanism," in *Proc. Int. Conf. Big Data, Inf. Comput. Netw. (BDICN)*, 2022, pp. 80–87, doi: [10.1109/BDICN55575.2022.00023](https://doi.org/10.1109/BDICN55575.2022.00023).
- [38] W. Yu, J. Su, and W. Zhang, "Research on short-term traffic flow prediction based on wavelet de-noising preprocessing," in *Proc. 9th Int. Conf. Natural Comput. (ICNC)*, 2013, pp. 252–256, doi: [10.1109/ICNC.2013.6817980](https://doi.org/10.1109/ICNC.2013.6817980).
- [39] K. L. Li, C. J. Zhai, and J. M. Xu, "Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN," in *Proc. Chin. Autom. Congr. (CAC)*, 2017, pp. 2804–2807, doi: [10.1109/CAC.2017.8243253](https://doi.org/10.1109/CAC.2017.8243253).



WUXIAO ZHOU was born in 1997. He received the master's degree from Chang'an University, under the supervision of an Associate Professor Xingli Jia. His major was Traffic and Transportation Engineering. He is currently working at Jiangsu Shagang Group Company Ltd.



SHUANGQING LI was born in 1997. He is currently pursuing the master's degree with Chang'an University, under the supervision of an Associate Professor Xingli Jia. His major is Traffic and Transportation Engineering.



XINGLI JIA received the Ph.D. degree in highway and railway engineering from Chang'an University. He is currently an Associate Professor of traffic and transportation engineering at Chang'an University. His research interests include road alignment and geometric design, traffic safety and emergency rescue, road ecological protection and landscape design, transportation GIS, and big data.



XINGPENG CHEN was born in 1997. He is currently pursuing the master's degree with Chang'an University, under the supervision of an Associate Professor Xingli Jia. His major is Traffic and Transportation Engineering.

• • •