Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 2265000, 20 pages
https://doi.org/10.1155/2022/2265000

*Research Article*

# An Improved CEEMDAN-FE-TCN Model for Highway Traffic Flow Prediction

**Heyao Gao, Hongfei Jia, and Lili Yang**

*School of Transportation, Jilin University, 5988 Renmin Street, Changchun 130022, China*

Correspondence should be addressed to Hongfei Jia; jiahf@jlu.edu.cn

Received 23 December 2021; Revised 3 March 2022; Accepted 18 April 2022; Published 10 May 2022

Academic Editor: Ruimin Li



With the advent of the data-driven era, deep learning approaches have been gradually introduced to short-term traffic flow prediction, which plays a vital role in the Intelligent Transportation System (ITS). A hybrid predicting model based on deep learning is proposed in this paper, including three steps. Firstly, an improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) method is applied to decompose the nonlinear time series of highway traffic flow to obtain the intrinsic mode function (IMF). The fuzzy entropy (FE) is then calculated to recombine subsequences, highlighting traffic flow dynamics in different frequencies and improving prediction efficiency. Finally, the Temporal Convolutional Network (TCN) is adopted to predict the recombined subsequences, and the final prediction result is reconstructed. Two sensors of US101-S on the main road and on-ramp were selected to measure the prediction effect. The results show that the prediction error of the proposed model on two sensors is notably decreased on single-step and multistep prediction, compared with the original TCN model. Furthermore, the proposed improved CEEMDAN-FE-X framework can be combined with prevailing prediction methods to increase the prediction accuracy, among which the improved CEEMDAN-FE-TCN model has the best performance and strong robustness.

## 1. Introduction

With the development of the social economy, the existing transportation supply has gradually been unable to meet the increasing traffic demand. Urban traffic congestion is continuously aggravated, resulting in economic losses, environmental pollution, and energy waste [1]. Intelligent Transport System (ITS), as an essential part of traffic management, combines the advanced technology of communication, information, and artificial intelligence. ITS aims to deliver real-time traffic information accurately to help travelers better route planning. At the same time, it can also improve the identification ability of traffic evolution trends and particular traffic situations, support the traffic management department to give early warning and command of emergencies, and effectively reduce casualties and economic losses [2–4]. Specifically, one of the critical technologies of ITS is short-term traffic prediction, which is the core of the active control of urban traffic systems [5]. Through the deep excavation of big data, the inherent evolution law of traffic flow can be mastered to achieve accurate and real-time prediction, which provides precise travel information for travelers and policy suggestions for managers to control beforehand. Traffic prediction of different periods has its application value. Short-term traffic flow prediction is essential for the traffic control department and travelers. For the traffic management department, short-term traffic flow prediction can help identify the evolution of traffic flow to formulate short-term traffic control measures such as lane closure and ramp control in advance, effectively alleviating potential traffic congestion. Besides, it can help travelers better understand the operation condition of the road network and make path planning accordingly [6–9]. Therefore, short-term traffic flow prediction is of practical significance and worth studying.

Two methods have dominated traffic forecasting research in the existing literature: statistical methods and machine learning methods [10]. Statistical methods based on linear statistics include the ARIMA method [11], Kalman filter method [12], Markov chain method, etc. [13], which is

more suitable for the road section with stable traffic conditions. However, the traffic flow nonlinearity is prominent when the prediction interval becomes smaller, resulting in low accuracy. Because of the fluctuation of traffic flow, the prediction method based on machine learning has drawn increasing attention, through which the inherent law of traffic data is excavated to capture the dynamics of traffic flow. For example, Wu et al. applied Support Vector Regression (SVR) to predict travel time and mapped the data to a high-dimensional space for regression, achieving good prediction results [14]. The SVR model is robust to noisy data and is more suitable for a small sample size. Cai et al. introduced K-Nearest Neighbor (KNN) model to realize multistep prediction on space and time, but the time complexity of calculation was high [15]. Besides, Csikos et al. constructed Artificial Neural Network (ANN) to learn the traffic speed dynamics through traffic speed samples in a month for prediction [16]. In recent years, with big data acquisition, deep learning models can capture more complex traffic features and have prospective applications [17, 18]. As one of the most typical methods, the Recurrent Neural Network (RNN) has a circular structure different from ANN. By feeding back the hidden layer information of the last moment to the input of the current moment, the temporal correlation of the traffic flow can be captured [19]. Traditional RNN mainly includes three structures: Elman Neural Network [20], Time-Delay Neural Network (TDNN) [21], and Nonlinear Autoregressive with Exogenous inputs Neural Network (NARX NN) [22]. Unfortunately, they all result in gradient vanishing and explosion problems, making it challenging to capture long-term information.

Nevertheless, it is shown that the traffic events that occurred in the previous period usually impact the predicted time, so RNN forecasting methods need to be further improved. Ma et al. firstly applied Long Short-Term Memory Neuron Network (LSTM NN) to predict traffic speed, which realized the memory of helpful information in a short and long time through the gate units and overcame the defect of traditional RNN. The results showed that the prediction performance was significantly better than other prevailing methods [23]. As the variant of LSTM, Gated Recurrent Unit (GRU) simplifies the structure, improving the prediction efficiency. Gao et al. combined GRU with MFD to forecast the traffic speed [24]. Other improved models like Attention-Based LSTM [25, 26] and BiLSTM [27, 28] achieved high accuracy on traffic prediction.

However, the process of RNN models is serial, meaning that later timesteps must wait for their predecessors to complete. For long-term sequence features capturing, RNNs use up much memory to store the partial results for their multiple cell gates. Convolution Neural Network (CNN) can extract the information of the long-term sequence parallelly because of the shared weights of the kernel [29]. As the length of the sequence increases, the network is deepened to learn the features, making it challenging to train. With the causal and dilated convolution, Temporal Convolution Network (TCN) achieves a flexible receptive field size, capturing the long-term historical information by a simple structure [30]. Zhao et al. improved the residual block of TCN for faster training speed and applied it to traffic flow prediction [31]. Zhang et al. used the genetic algorithm to optimize the hyperparameters of TCN. The results showed that the prediction performance was significantly better than other prevailing methods [32].

The inherent changing law of traffic flow is complex, consisting of various dynamics on different temporal scales. Although the deep learning models can capture long-term historical information, they need a deep network and take up much training time and memory. Thus, it is necessary to decompose the traffic flow time series, which simplifies the structure of prediction models and extracts features thoroughly and effectively. Huang et al. proposed Empirical Mode Decomposition (EMD), which decomposed the trend or fluctuation of different scales in signals consecutively to generate a series of IMF with different frequencies [33, 34]. Unlike wavelet transform, it is an adaptive and data-driven method without a defined wavelet basis. Theoretically, signals with nonlinearity and randomness can be decomposed. However, the conventional EMD decomposes the signal incompletely, causing mixing and false modes. Thus, several improved models were proposed to solve these problems [35–38]. In recent years, EMD related methods have been gradually introduced to traffic prediction. For instance, Wei et al. combined EMD with Backpropagation Neural Network (BPNN) to predict the subway passenger flow, which showed notable performance. Only modes highly correlated with the original data were selected to improve the prediction efficiency [39]. Likewise, Chen et al. applied Ensemble Empirical Mode Decomposition (EEMD) to decompose the traffic flow time series, removed the high-frequency mode, and introduced LSTM NN to predict the left reconstructed modes [40]. However, what cannot be ignored is that each IMF plays an essential part in the time series, and the immediate abandonment of some modes will lead to the lack of detailed information on traffic flow features. Lu et al. applied the XGBoost method to predict the traffic flow intrinsic mode function (IMF) of each lane after Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) [41]. Wang et al. combined CEEMDAN with LSSVM to predict highway traffic flow [42]. Huang et al. introduced K-means to cluster the traffic flow IMF decomposed by CEEMDAN and predicted by BiLSTM [43]. However, the value of K has not been chosen with a theoretical basis, and the BiLSTM may take up much memory usage. Though mixing modes were solved to some extent, the residual noise and spurious modes remained. Also, the prediction on every IMF resulted in poor efficiency. Moreover, the in-depth change features of traffic flow may not be captured because of the small training data size or high memory usage.

The existing research on the decomposition prediction of traffic flow time series is insufficient and remains preliminary. Such problems as incomplete decomposition, low prediction efficiency, high storage of memory, and the deep capture of traffic flow dynamics need further investigation. Therefore, in this paper, an improved CEEMDAN-FE-TCN model is proposed to forecast highway traffic flow. First, the improved CEEMDAN method decomposes the nonlinear

highway traffic flow into IMF and residual with different frequencies. Next, the fuzzy entropy (FE) of each mode is calculated. IMF and residual with similar chaos are recombined, highlighting the traffic dynamics. Finally, the TCN is applied to predict the different recombined subsequences. After reconstructing the output of TCN submodels, the predicted traffic flow is obtained. The contributions of the paper can be summarized as follows:

(i) The improved CEEMDAN method is first used for highway traffic flow decomposition. The changing features are decomposed to different temporal scales, making TCN extract the dynamics thoroughly.

(ii) The FE difference of different modes decomposed from the original data is calculated. On this basis, the modes are recombined as subsequences, which highlights the primary trend of traffic flow changes and retains specific fluctuations. The computational complexity is reduced, and the forecasting efficiency and accuracy are further improved.

(iii) The proposed improved CEEMDAN-FE-X framework can be applied to decrease the prediction error of prevailing models notably. Moreover, the improved CEEMDAN-FE-TCN model outperforms other models compared in this paper, which has strong robustness.

The rest of the paper is arranged as follows: In section 2, an improved CEEMDAN-FE-TCN model is proposed for traffic flow prediction. Section 3, Section 4, and Section 5 introduce the improved CEEMDAN, FE, and TCN, respectively. The prediction effects of the proposed model are verified on two sensors in Section 6. Finally, Section 7 summarizes the conclusions and future directions.

## 2. The Improved CEEMDAN-FE-TCN Model

In this paper, an improved CEEMDAN-FE-TCN model is constructed for highway traffic flow prediction, which contains three modules: improved CEEMDAN decomposition, FE calculation, and TCN prediction.

TCN is applied as the core module to predict the highway traffic flow. As a new neural network with a convolutional structure, TCN has the advantages of large-scale parallel processing of CNN and integrates the modeling ability of sequential tasks, which makes up for the long-term dependence problem of RNN [44]. The RNN variants like LSTM and GRU memorize part of the information through the gated unit, while TCN can capture all the historical information with better prediction and faster training speed [30].

However, the traffic flow time series consists of different temporal scaled changing features, causing fluctuation and nonlinearity. It is challenging for TCN to extract the mixed dynamics thoroughly. So, the improved CEEMDAN model is adopted to decompose the sequence to IMF and residual, making TCN capable of capturing the features on every single temporal scale.

The modes decomposed by improved CEEMDAN have physical significance. Nevertheless, from the traffic point of view, some IMF may be part of traffic flow dynamics on a specific time scale. Besides, each IMF needs a corresponding TCN submodel for training and predicting, causing complex computation. Therefore, FE is introduced to calculate the complexity of every IMF decomposed by the traffic flow time series. The sequences with close FE have similar temporal scales and stationarity, indicating that TCN will have the same feature extracting ability on the recombined sequence as every single sequence. The recombination will highlight the changing features of traffic flow and eliminate the accumulated error on multiple similar sequences prediction. Thus, the modes with similar FE are recombined as the input of TCN, reducing calculation complexity and improving prediction efficiency and accuracy.

The output of every TCN submodel is the predicted traffic flow on different time scales. After reconstruction, the final predicted traffic flow is obtained.

The framework of the proposed model is shown in Figure 1.

The procedures in specific are expressed as follows:

Step 1: the improved CEEMDAN method is introduced to decompose the original traffic flow time series to obtain $k$ IMF and residual with different frequencies.

Step 2: the FE of each mode is calculated. According to the difference between the modes, the IMF and residual with similar chaos are recombined to subsequences (RS).

Step 3: the TCN submodules are adopted to train and predict RS (1)-RS ($n$), respectively; then the prediction results are reconstructed to obtain the predicted highway traffic flow.

## 3. Improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise

*3.1. CEEMDAN Algorithm.* The CEEMDAN algorithm can eliminate the mixing modes to some extent. Each IMF is calculated through the residual signal by adding white noise adaptively in the IMF decomposition process, reducing the reconstruction error. The method has good integrity and reduces the number of integrations. The specific steps are shown as follows [37]:

Step 1: a series of Gaussian white noise is added adaptively to the original signal $x$:

$$x^{(i)} = x + \beta_0 \omega^{(i)}, \quad i = 1, \ldots, I. \tag{1}$$

$x^{(i)}$ denotes the time series after adding white noise for the $i$th time; $\beta_0$ denotes the noise factor; $\omega^{(i)}$ denotes the white noise added for the $i$th time; $I$ denotes the number of integrations.

Step 2: the EMD algorithm is used to decompose $x^{(i)}$, and the first EMD mode $d_1^{(i)}$ is averaged to calculate the first CEEMDAN mode as follows:

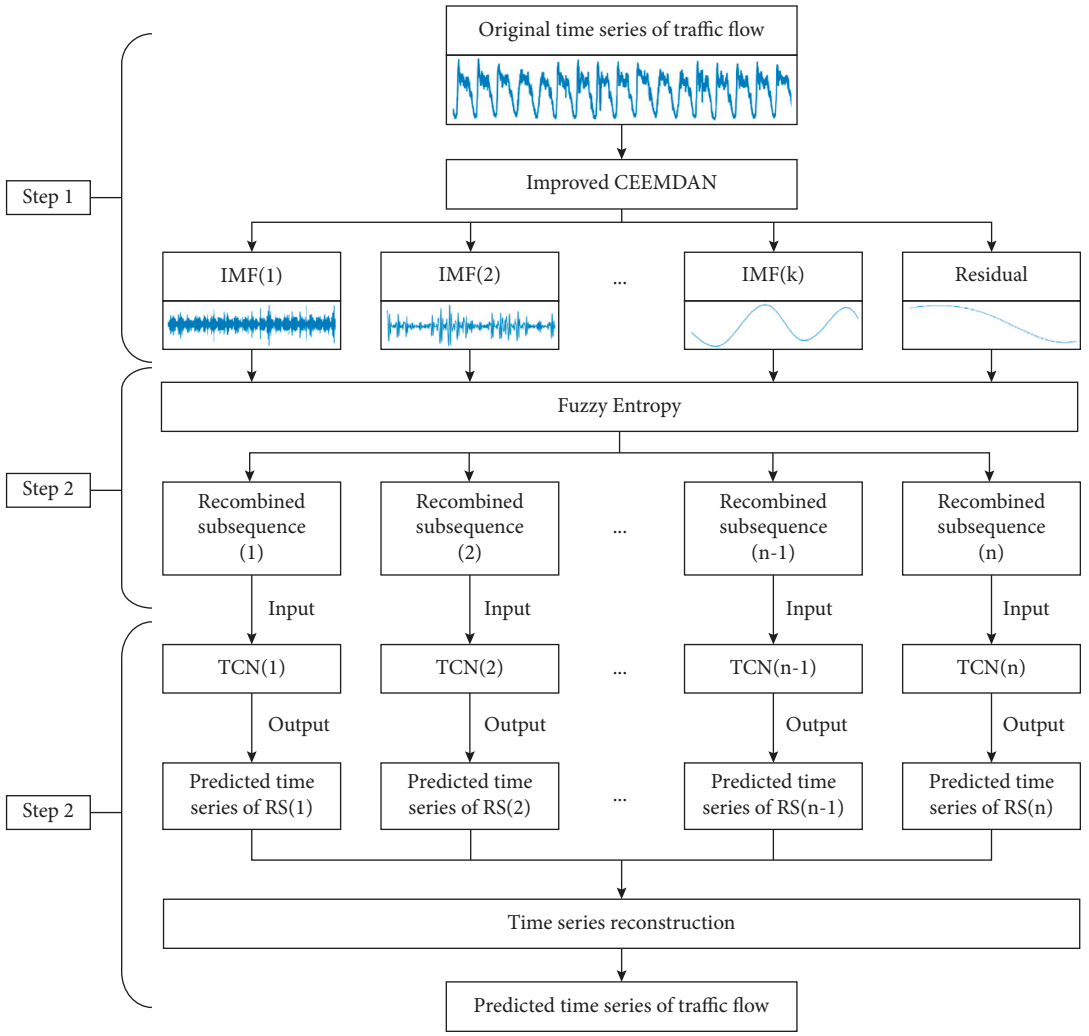


Figure 1: The framework of the proposed model.

$$\tilde{d}_1 = \frac{1}{I}\sum_{i=1}^{I} d_1^{(i)}. \tag{2}$$

Remove $\tilde{d}_1$ from $x$ to obtain the first residue as in

$$r_1 = x - \tilde{d}_1. \tag{3}$$

Step 3: decompose $r_1 + \beta_1 E_1(\omega^{(i)})$ by the EMD algorithm to obtain the second CEEMDAN mode:

$$\tilde{d}_2 = \frac{1}{I}\sum_{i=1}^{I} E_1\left(r_1 + \beta_1 E_1\left(\omega^{(i)}\right)\right), \tag{4}$$

where $E_k(\cdot)$ denotes the $k$th mode decomposed by the EMD algorithm.

Step 4: repeat the following process to calculate the remaining modes until the remaining residual cannot decompose.

$$\begin{aligned} r_k &= r_{(k-1)} - \tilde{d}_k, k = 2, \ldots, K, \\ \tilde{d}_{(k+1)} &= \frac{1}{I}\sum_{i=1}^{I} E_1\left(r_k + \beta_k E_k\left(\omega^{(i)}\right)\right), \end{aligned} \tag{5}$$

where $K$ denotes the number of the CEEMDAN modes. The final residual is calculated as

$$r_K = x - \sum_{k=1}^{K} \tilde{d}_k. \tag{6}$$

The original $x$ can be expressed as

$$x = \sum_{k=1}^{K} \tilde{d}_k + r_K. \tag{7}$$

*3.2. Improvements on CEEMDAN.* Although the CEEMDAN method has overcome mode mixing, residual noise and spurious modes remain. On this basis, the improved CEEMDAN algorithm was proposed, which has two perfections: One is to estimate the local mean of the signal plus noise and define the difference between the current residue and the average of its local means as the primary mode, which reduces the residual noise existing in the decomposition mode. The other is to extract the $k$th mode by using $E_k(\omega^{(i)})$ to replace white noise, reducing mode overlap. Therefore, the improved CEEMDAN method is adopted to decompose the original traffic flow time series. The steps can be described as follows [38]:

Define operator $E_k(\cdot)$ as the $k$th mode decomposed by EMD, operator $M(\cdot)$ as the local mean of the mode, and operator $\cdot$ as mean operation. Then, $E_1(x) = x - M(x)$.

Step 1: $x^{(i)} = x + \beta_0 E_1(\omega^{(i)})$ is constructed to calculate the first residue:

$$r_1 = M\left(x^{(i)}\right). \tag{8}$$

Step 2: the first mode can be calculated as

$$\widetilde{d}_1 = x - r_1. \tag{9}$$

Step 3: the second residue is estimated as the mean of a series of $r_1 + \beta_1 E_2(\omega^{(i)})$ and the second mode is defined as

$$\begin{aligned}\widetilde{d}_2 &= r_1 - r_2 \\ &= r_1 - M\left(r_1 + \beta_1 E_2\left(\omega^{(i)}\right)\right).\end{aligned} \tag{10}$$

Step 4: for $k = 3, \ldots, K$, the $k$th residue is expressed as

$$r_k = M\left(r_{k-1} + \beta_{k-1} E_k\left(\omega^{(i)}\right)\right). \tag{11}$$

Step 5: the $k$th mode of the improved CEEMDAN can be obtained:

$$\widetilde{d}_k = r_{k-1} - r_k. \tag{12}$$

Step 6: go to Step 4 for next $k$.

## 4. Fuzzy Entropy

Fuzzy entropy (FE) measures the complexity of time series and the probability of generating new patterns when the dimension changes. The higher the time series complexity, the higher the entropy [45]. The fuzzy membership function is introduced to make the fuzzy entropy continuous and smooth with the change of parameters, reducing the sensitivity dependence on parameters, and the statistical results are stable [46]. The process of FE calculation is shown as follows [47]:

Step 1: the dimension is set for the IMF of traffic flow time series $X = [x(1), x(2), \ldots, x(N)]$, and the $m$-dimension vector is constructed as follows:

$$X_m(i) = \begin{bmatrix} x(i), x(i+1), \ldots, \\ x(i+m-1) \end{bmatrix} - u(i). \tag{13}$$

$i = 1, 2, \ldots, N - m + 1$; then, $u(i)$ can be expressed as

$$u(i) = \frac{1}{m} \sum_{j=0}^{m-1} x(i+j). \tag{14}$$

Step 2: the distance $d_{ij}^m$ of vectors $X_m(i)$ and $X_m(j)$ is calculated as

$$d_{ij}^m = \max_{k=1,\ldots,m-1} \left( \begin{vmatrix} (x(i+k) - u(i)) \\ -(x(j+k) - u(j)) \end{vmatrix} \right). \tag{15}$$

$i, j = 1, 2, \ldots, N - m + 1$, and $i \neq j$.

Step 3: introduce the membership function:

$$A(x) = \begin{cases} 1, & x = 0, \\ \exp\left[-\ln(2)\left(\dfrac{x}{r}\right)^2\right], & x > 0. \end{cases} \tag{16}$$

$r$ denotes the similarity tolerance parameter, which means $R$ times the standard deviation of the original one-dimensional time series, namely, $r = R \times SD$.

The similarity between vectors $X_m(i)$ and $X_m(j)$ is defined as

$$A_{ij}^m = \begin{cases} 1, & d_{ij}^m = 0, \\ \exp\left[-\ln(2)\left(\dfrac{d_{ij}^m}{r}\right)^2\right], & d_{ij}^m > 0. \end{cases} \tag{17}$$

Step 4: define function

$$C_i^m(r) = \frac{1}{N-m} \sum_{j=1, j\neq i}^{N-m+1} A_{ij}^m. \tag{18}$$

Then,

$$\phi^m(r) = \frac{1}{N-m+1} \sum_{i=1}^{N-m+1} C_i^m. \tag{19}$$

Step 5: go to Step 1 for next $m$.

Step 6: the fuzzy entropy of traffic flow time series can be expressed as

$$\text{Fuzzy}En(m, r, N) = \ln \phi^m(r) - \ln \phi^{m+1}(r). \tag{20}$$

## 5. Temporal Convolutional Network

TCN combines the advantages of CNN and RNN, which capture the global information and process parallelly. It contains three main modules: causal convolution, dilated convolution, and residual block.

*5.1. Causal Convolution.* When processing sequential tasks, TCN needs to generate outputs with the same length as the input. All data in causal convolution strictly follow the causal relationship in time order, meaning that the value at time $t$ only depends on the information before time $t$. Because of the strict time-constrained nature of causal convolution, TCN ensures causality and prevents future data leakage.

*5.2. Dilated Convolution.* With the increasing length of the sequence, the network is deepened to extract more features of historical time, making it hard to train. In order to simplify the network structure, the dilated convolution is adopted, which enables an exponentially sizeable receptive field. For a 1D sequence input $x \in R$ and a filter $f$: $\{0, \ldots, k-1\} \longrightarrow R$, the traffic flow $F$ at time $s$ is defined as

$$F(s) - (x *_d f)(t) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i}, \tag{21}$$

where $d$ is the dilation factor, $k$ is the filter size, and $s-d \cdot i$ indicates the past direction. The structure of the causal and dilated convolution is shown in Figure 2. With the dilated convolution, the receptive field size of TCN is flexible, making it easy to capture the features of the global long sequence by a few hidden layers.

*5.3. Residual Block.* By learning the identity mapping function, residual connection enables the network to transfer information in a cross-layer way, increasing network depth, improving accuracy, and simplifying network training.

$X$ is set as the input value of the residual module, and the potential identity mapping function for cross-layer is $F(\cdot)$, the result of which will be added to the input value $X$, so the output value $o$ of the residual module can be expressed as

$$o = \text{Activation}(X + F(X)). \tag{22}$$

The structure of a residual block is shown in Figure 3.

## 6. Empirical Study

*6.1. Data Description.* In this paper, two sections on US101-S in California were selected as examples to verify the effectiveness of the proposed model. The section where VDS No. 717490 locates is on the mainline, and the section where VDS No. 718462 locates is on the on-ramp. The locations are shown in Figure 4, and the detailed information of the two sensors is shown in Table 1.

The datasets were collected by Caltrans PeMS (https://pems.dot.ca.gov/) from 2018/8/1 to 2018/8/31. The flow of all lanes was aggregated into 5-minute intervals to reduce the volatility of the data and ensure real-time prediction. There were 8928 samples in each group of datasets. The error and loss rate was less than 2%, making it proper to be trained and tested. The training and testing datasets were divided by 2018/8/27. There were 7488 samples trained and 1440 samples tested in each group, as shown in Figure 5.

The autocorrelation of the traffic flow data obtained by VDS No. 717490 and VDS No. 718462 is shown in Figure 6. As the time lag increases, the autocorrelation of both sequences decreases slowly. Therefore, they are nonstationary time series with nonlinear changes which should be smoothed. In addition, when the time lag increases to 40, the autocorrelation is still over 0.3, indicating that the sequences have a long-time dependence, so TCN is suitable for the traffic flow prediction.

The datasets were processed by TensorFlow2.0.0 and Keras 2.3.1 and compiled by Python3.6. Four indexes were introduced to measure the prediction accuracy: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R-squared, and Geoffrey E. Havers (GEH). They are calculated as follows:

$$\begin{aligned} \text{MAE} &= \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \\ \text{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|^2}, \\ R^2 &= 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y}_i)^2}, \\ \text{GEH} &= \sqrt{\frac{2(y_i - \hat{y}_i)^2}{y_i + \hat{y}_i}}. \end{aligned} \tag{23}$$

*6.2. Traffic Flow Sequence Decomposition and Recombination.* The improved CEEMDAN algorithm was adopted to decompose the traffic flow time series obtained by VDS No. 717490 and VDS No. 718492, respectively. The 11 IMF and one residual were arranged with different frequencies, as shown in Figure 7.

The IMF and residual with similar FE were recombined to reduce the calculation complexity and increase the forecasting efficiency and accuracy. The mode FE and the FE difference between different modes of the traffic flow sequences obtained from the two sensors were calculated, as shown in Table 2, and the changing trend of FE is shown in Figure 8. For VDS No. 717490, modes with FE differences less than 0.1 were recombined. Similarly, for VDS No. 718462, 0.05 was the difference threshold of recombination. The recombined subsequences of the two sensors are plotted in Figure 9.

Figure 9 shows that each recombined subsequence reflects part of traffic flow dynamics. For VDS No. 717490, IMF1, IMF2, and IMF3 are high-frequency modes with high FE and chaos. Although they are noise with poor predictability, reflecting the randomness and nonlinearity of traffic flow, the detailed information is contained. Therefore, they need to be predicted, respectively.

The FE of IMF4 is close to that of IMF5 and IMF6. The recombined subsequence reflects the specific daily change characteristics of traffic flow. There are two peaks every 288

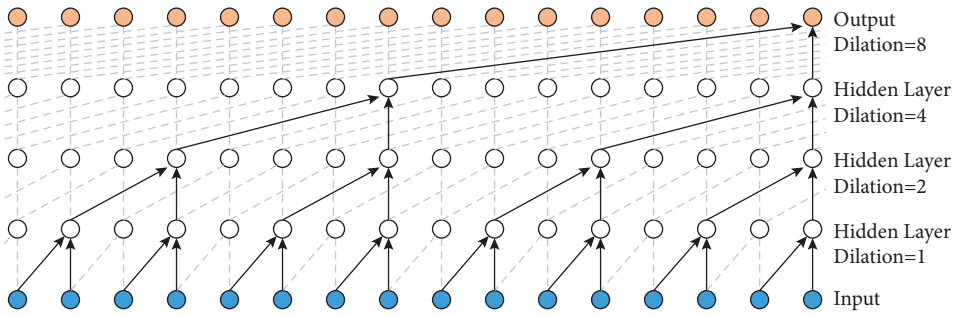


Figure 2: A dilated causal convolution with dilation factors $d = 1, 2, 4, 8$ and filter size $k = 2$ [48].

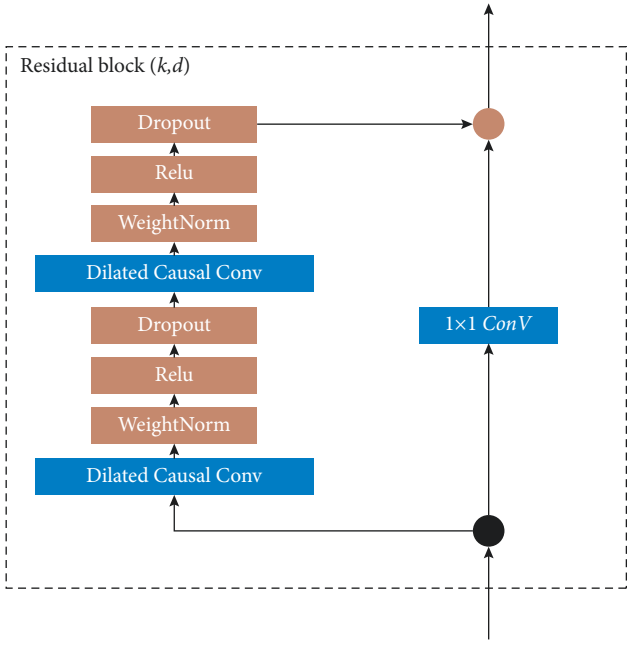


Figure 3: TCN residual block.

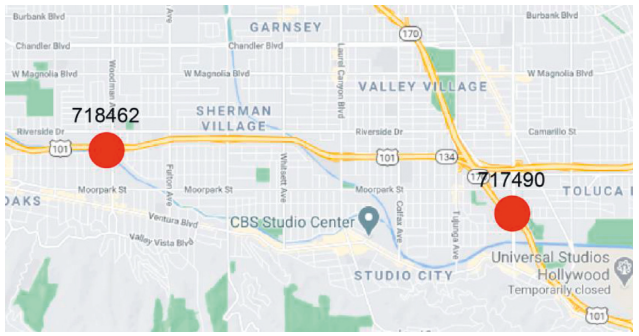


Figure 4: The locations of VDS No. 718462 and VDS No. 717490 for testing.

data points, representing the morning and evening peaks of traffic flow which have apparent differences. It is shown that the morning peak flow is higher than that of the evening on weekdays, while on weekends, the two peaks are similar and lower than those on weekdays.

The FE of IMF7 is quite different from IMF6 and IMF8, reflecting that the overall trend of daily traffic flow increases first and then decreases. Together with IMF4, IMF5, and IMF6, they are median-frequency modes with solid predictability and are the core of time series prediction.

IMF8, IMF9, IMF10, and IMF11, together with the residual, constitute the trend mode, reflecting the weekly traffic flow dynamics. It is shown that the traffic flow on weekdays is relatively stable and higher than that on weekends. The FE and the chaos of the trend mode are low, and the predictability is firm. The trend mode is the essential component of time series prediction. It is worth noting that the data on 2018/8/21 were unstable and fluctuant, so the corresponding subsequences of IMF7 and IMF8 + IMF9 + IMF10 + IMF11 + Residual changed apparently on that day, causing disturbance to the original changing cycle.

The traffic flow obtained from VDS No. 718462 shows similar changing characteristics to VDS No. 717490. However, because the on-ramp only has one lane with more unstable traffic flow, the fluctuation frequency is higher than that of the mainline, resulting in weaker periodicity.

Table 1: Detailed information on the two sensors.

| Sensor ID | Freeway | Sensor position | Lane number |
|---|---|---|---|
| 717490 | US101-S | Mainline | 5 |
| 718462 | US101-S | On ramp | 1 |

### 6.3. Highway Traffic Flow Prediction

*6.3.1. Hyperparameter Optimization.* The accuracy of each prediction model is affected by various hyperparameters, which should be optimized before the prediction. For TCN, the number of filters, time lag, kernel size, and dilation factors are the crucial hyperparameters affecting the performance. The number of filters determines whether feature extraction is complete, the others affect the size of the receptive field, and all hyperparameters jointly influence the prediction accuracy of TCN. GridSearchCV in the Scikit-learn was imported to score the performance of different hyperparameters combinations of each prediction model and search for the best hyperparameters by 10-fold cross-validation. The data range of different hyperparameters of each TCN module is shown in Table 3.

*6.3.2. Results and Comparison.* The prediction effect can be divided into vertical and horizontal comparisons. The

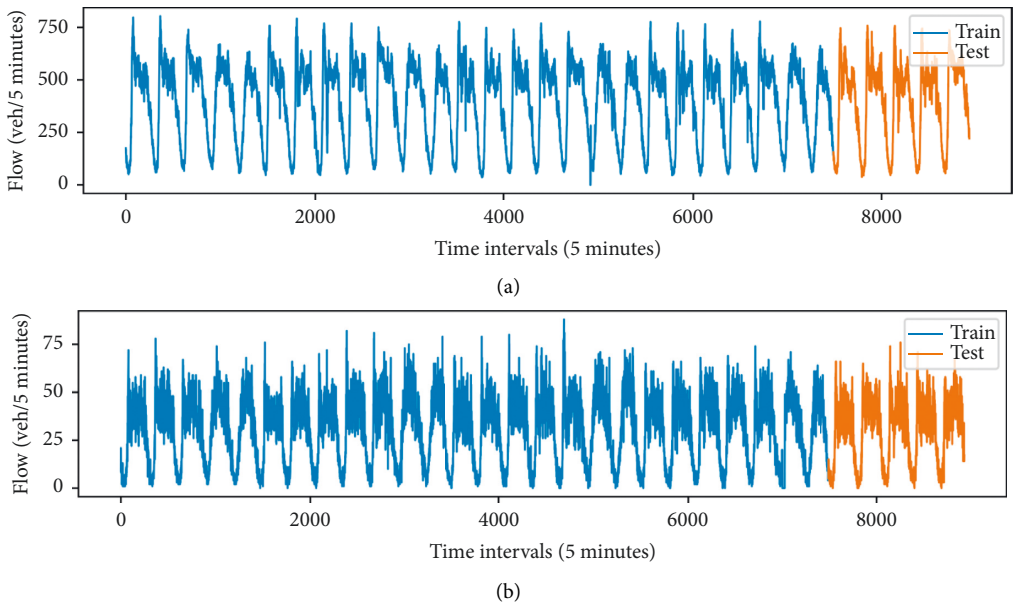


Figure 5: Training and testing datasets of (a) 717490 and (b) 718462.

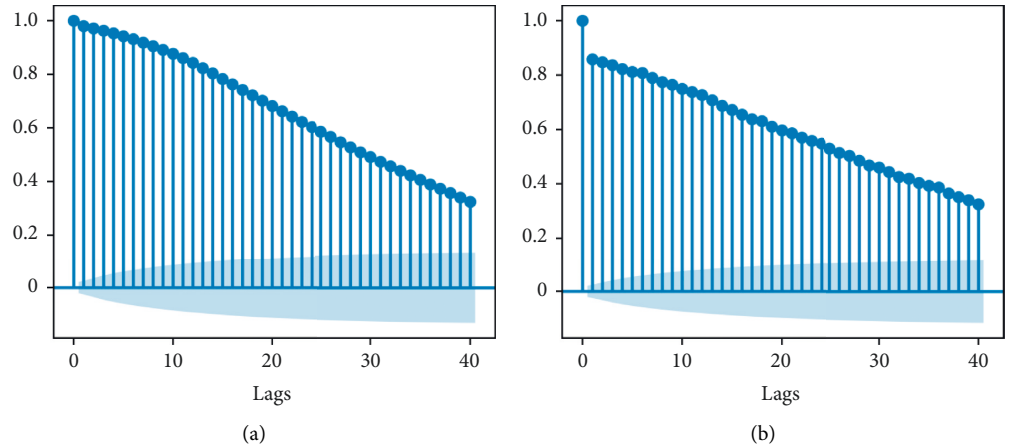


Figure 6: Autocorrelation of (a) 717490 and (b) 718462.

vertical comparison measures the effect of different variants of the TCN model, while horizontal comparison compares different baseline models for traffic flow prediction. Both comparisons were analyzed to verify the superiority of the proposed model. In addition, our experiment platform is a personal computer with Core (TM) i3-8100 CPU@3.60 GHz and 8 GB RAM. Python 3.6, TensorFlow 2.0.0, and Keras 2.3.1 are used to realize the models.

*(1) Vertical Comparison of Different Models Based on TCN.* The modes and the recombination subsequences decomposed by VDS No. 717490 and VDS No. 718462 traffic flow time series were predicted. The error and training time is shown in Tables 4 and 5, respectively.

The results show that the recombined subsequences have higher accuracy and less training time than the single IMF. With recombination, the number of training models is reduced, and the computational complexity is decreased. Despite the improved CEEMDAN algorithm, other methods based on EMD are introduced to optimize the prediction performance of TCN. The results are shown in Table 6 and Figure 10.

As shown in Figure 10, compared with the direct prediction, the accuracy of decomposition prediction is notably increased. With the improvement of EMD, the performance of forecast is promoted. Specifically, for VDS No. 717490, the error of ICEEMDAN-TCN is reduced by 69% compared with TCN. For VDS No. 718462, it is decreased by 59%. Furthermore, recombining similar modes according to FE can ulteriorly improve efficiency and accuracy with less calculation complexity. In terms of prediction efficiency, the training time of the model after the recombination is reduced by 34%–49%. From the perspective of prediction accuracy, for VDS No. 717490, the error of ICEEMDAN-FE-TCN is further reduced by 3% compared with ICEEMDAN-TCN, and for VDS No. 718462, it is decreased by 5%.

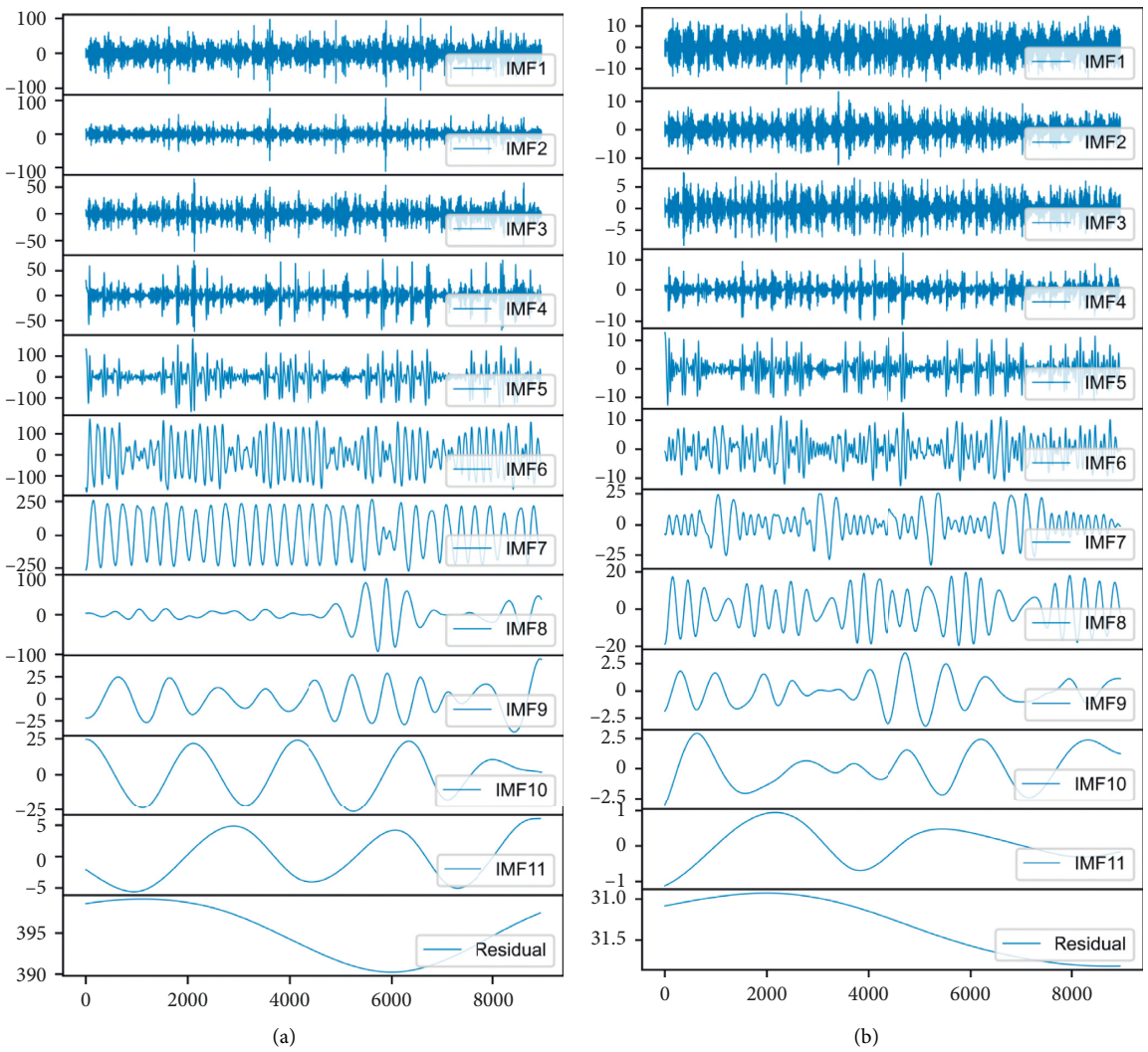


Figure 7: IMF and residual of (a) 717490 and (b) 718462.

Table 2: IMF FE and the FE difference of 717490 and 718462.

| | 717490 | | | 718462 | |
|---|---|---|---|---|---|
| $i$ | $FE_i$ | $FE_{i+1}-FE_i$ | $i$ | $FE_i$ | $FE_{i+1}-FE_i$ |
| 1 | 1.5412 | 0.4153 | 1 | 0.9249 | 0.3501 |
| 2 | 1.1259 | 0.5193 | 2 | 0.5748 | 0.2381 |
| 3 | 0.6066 | 0.2099 | 3 | 0.3367 | 0.0357 |
| 4 | 0.3967 | 0.0541 | 4 | 0.3010 | 0.0509 |
| 5 | 0.3426 | 0.0097 | 5 | 0.2501 | 0.0250 |
| 6 | 0.3329 | 0.1236 | 6 | 0.2251 | 0.0717 |
| 7 | 0.2093 | 0.1627 | 7 | 0.1534 | 0.0352 |
| 8 | 0.0466 | −0.0075 | 8 | 0.1182 | 0.1039 |
| 9 | 0.0541 | 0.0363 | 9 | 0.0143 | 0.0073 |
| 10 | 0.0178 | 0.0128 | 10 | 0.0070 | 0.0054 |
| 11 | 0.0050 | 0.0036 | 11 | 0.0016 | 0.0013 |
| 12 | 0.0014 | | 12 | 0.0002 | |

Overall, the proposed model has the lowest error and the least training time (except original TCN) on both sensors, indicating the best goodness of fit. Aug 31, 2018, was taken as an example to visualize the prediction performance of each model, as shown in Figure 11. Since there is little difference in visualization between X-TCN and X-FE-TCN, the X-FE-TCN is representative.

The prediction performance of the original TCN model is approximately fitted to the actual data but has an apparent time delay. The reason is that the traffic flow time series consists of changing features with multiple frequencies. TCN cannot accurately capture the multiple-scaled dynamics, causing prediction error. The EMD-based models can decompose the sequence to different IMF, making it easier for TCN to learn the characteristics of every scale so that the prediction performs better than the original model, and the hysteresis can be effectively eliminated. Among all the models compared, the improved CEEMDAN-FE-TCN achieves the best performance because of the extraordinary ability of decomposition. Besides, the fluctuation of ramp flow is more potent than that of the mainline flow; the proposed model also performs well on the ramp traffic flow prediction, which appears to have strong robustness.

*(2) Horizontal Comparison of Eight Different Models.* The traffic flow of VDS No. 717490 and VDS No. 718462 was

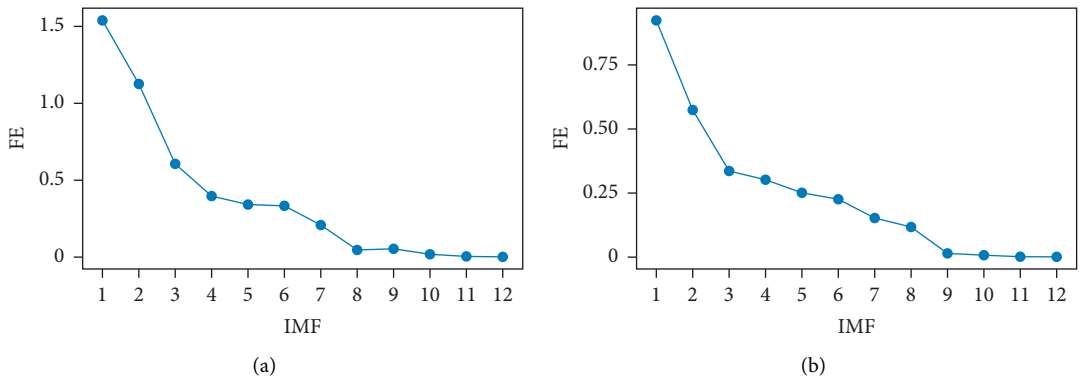


Figure 8: IMF FE of (a) 717490 and (b) 718462.

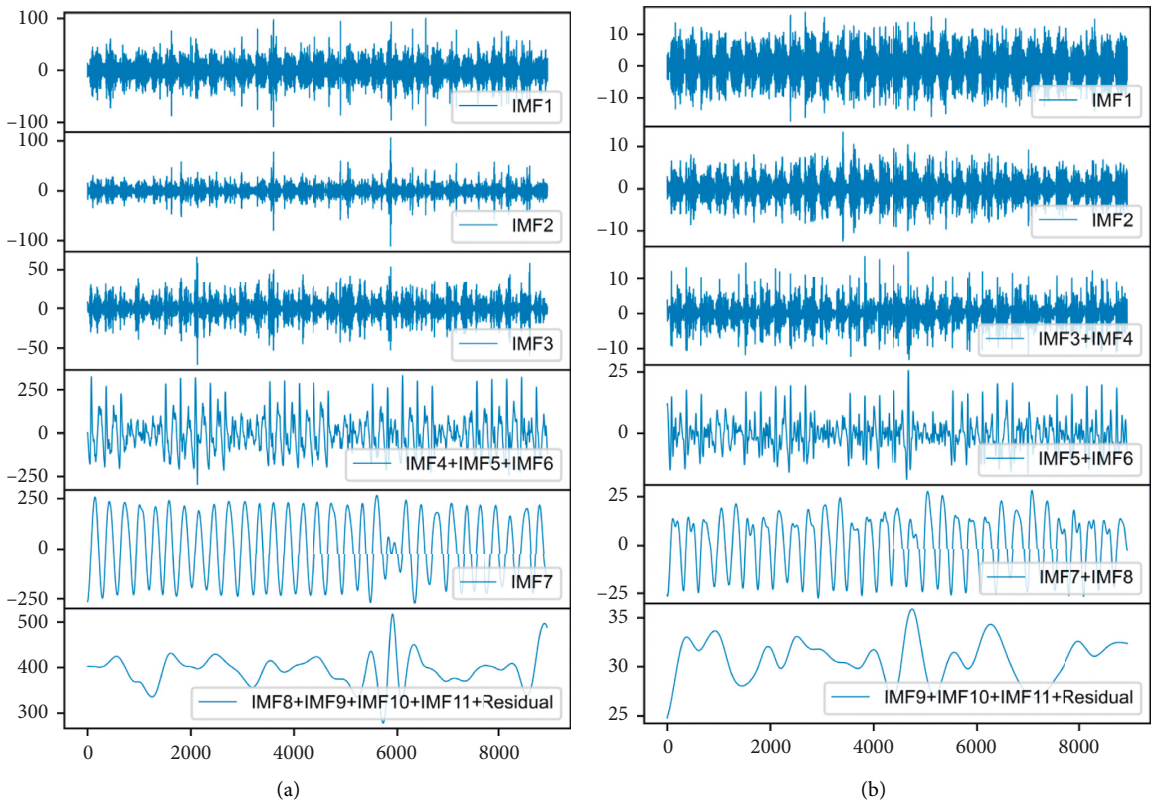


Figure 9: Recombined subsequences of (a) 717490 and (b) 718462.

Table 3: TCN hyperparameters.

| Hyperparameters | Range |
|---|---|
| *nb_filters* | 16, 32, 64, 128 |
| *time_lag* | 10, 20, 30, 40 |
| *kernel_size* | 2, 3, 4, 5 |
| *dilation_factors* | {1, 2}, {1, 2, 4}, {1, 2, 4, 8}, {1, 2, 4, 8, 16} |

predicted single-step and multistep ahead by TCN, LSTM, GRU, SVR, and HA algorithms and their improved models under the framework proposed in this paper. The hyperparameters optimization method was mentioned in 6.3.1. The prediction error of each model is shown in Table 7 and Figure 12. The visualization is shown in Figures 13 and 14.

The results above show that the traffic flow predicted single-step and multistep ahead by different algorithms approximately fits with the original data, and the prediction accuracy obtained by decomposition forecasting is significantly improved compared with the direct prediction.

From the perspective of one-step-ahead prediction, for VDS No. 717490, the prediction accuracy of TCN is higher than that of LSTM, GRU, SVR, and HA. Under the framework of the improved CEEMDAN-FE-X, the prediction error of TCN, LSTM, GRU, and SVR is sharply decreased by 69%, 64%, 67%, and 44%, respectively. Among all the models compared, the improved CEEMDAN-FE-TCN model obtains the lowest MAE, RMSE, GEH average at 7.36, 10.34, and 0.39, respectively, and the highest R-square

Table 4: Prediction error and training time of TCN in different IMF of 717490.

| IMF | MAE | RMSE | Training time (s) | IMF | MAE | RMSE | Training time (s) |
|---|---|---|---|---|---|---|---|
| 1 | 7.409 | 10.492 | 34.792 | 7 | 0.476 | 0.945 | 22.534 |
| 2 | 2.834 | 3.877 | 48.228 | 8 | 0.209 | 0.563 | 28.836 |
| 3 | 0.802 | 1.095 | 32.179 | 9 | 0.109 | 0.238 | 22.735 |
| 4 | 0.612 | 1.063 | 44.084 | 10 | 0.003 | 0.007 | 20.575 |
| 5 | 1.079 | 1.201 | 40.480 | 11 | 0.005 | 0.006 | 22.510 |
| 6 | 0.544 | 1.064 | 32.451 | 12 | 0.003 | 0.011 | 14.848 |
| 4 + 5 + 6 | 0.646 | 1.150 | 31.249 | 8 + 9 + 10 + 11 + 12 | 0.251 | 0.549 | 24.654 |

Table 5: Prediction error and training time of TCN in different IMF of 718462.

| IMF | MAE | RMSE | Training time (s) | IMF | MAE | RMSE | Training time (s) |
|---|---|---|---|---|---|---|---|
| 1 | 2.088 | 2.981 | 33.178 | 7 | 0.063 | 0.108 | 30.771 |
| 2 | 0.596 | 0.817 | 42.502 | 8 | 0.063 | 0.119 | 36.728 |
| 3 | 0.207 | 0.257 | 44.560 | 7 + 8 | 0.049 | 0.077 | 32.792 |
| 4 | 0.046 | 0.107 | 38.292 | 9 | 0.001 | 0.002 | 14.703 |
| 3 + 4 | 0.172 | 0.241 | 44.404 | 10 | 0.001 | 0.001 | 20.681 |
| 5 | 0.056 | 0.072 | 42.712 | 11 | 0.001 | 0.001 | 30.917 |
| 6 | 0.017 | 0.025 | 30.733 | 12 | 0.001 | 0.001 | 14.653 |
| 5 + 6 | 0.054 | 0.092 | 29.380 | 9 + 10 + 11 + 12 | 0.001 | 0.003 | 22.835 |

Table 6: Prediction error and training time of different models based on TCN of 717490 and 718462.

| Sensor ID | Model | MAE | RMSE | GEH (Average) | *R*-Square | Training time (s) |
|---|---|---|---|---|---|---|
| 717490 | TCN | 24.98 | 34.33 | 1.31 | 0.9668 | 96* |
| | EMD-TCN | 15.62 | 20.17 | 0.82 | 0.9885 | 450 |
| | EMD-FE-TCN | 15.33 | 19.96 | 0.80 | 0.9888 | 262 |
| | EEMD-TCN | 9.71 | 13.01 | 0.51 | 0.9952 | 467 |
| | EEMD-FE-TCN | 8.76 | 12.07 | 0.45 | 0.9959 | 241 |
| | CEEMDAN-TCN | 8.93 | 12.16 | 0.47 | 0.9958 | 413 |
| | CEEMDAN-FE-TCN | 8.12 | 11.31 | 0.42 | 0.9964 | 231 |
| | ICEEMDAN-TCN | 7.61 | 10.54 | 0.41 | 0.9969 | 364 |
| | ICEEMDAN-FE-TCN | 7.36* | 10.34* | 0.39* | 0.9970* | 194 |
| Sensor ID | Model | MAE | RMSE | GEH (Average) | R-square | Training time (s) |
| 718462 | TCN | 5.31 | 7.01 | 1.00 | 0.8203 | 71* |
| | EMD-TCN | 3.64 | 4.73 | 0.69 | 0.9180 | 455 |
| | EMD-FE-TCN | 3.63 | 4.72 | 0.69 | 0.9186 | 298 |
| | EEMD-TCN | 2.52 | 3.40 | 0.48 | 0.9577 | 464 |
| | EEMD-FE-TCN | 2.49 | 3.38 | 0.47 | 0.9583 | 262 |
| | CEEMDAN-TCN | 2.31 | 3.13 | 0.44 | 0.9642 | 446 |
| | CEEMDAN-FE-TCN | 2.28 | 3.10 | 0.43 | 0.9649 | 236 |
| | ICEEMDAN-TCN | 2.22 | 3.05 | 0.42 | 0.9660 | 380 |
| | ICEEMDAN-FE-TCN | 2.10* | 2.96* | 0.39* | 0.9681* | 205 |

*Note.* *indicates the best results; ICEEMDAN means improved CEEMDAN.

at 0.997. For VDS No. 718462, the prediction accuracy of TCN is higher than that of LSTM, GRU, SVR, and HA. Under the framework of the improved CEEMDAN-FE-X, the prediction error of TCN, LSTM, GRU, and SVR is sharply decreased by 59%, 55%, 54%, and 51%, respectively. Among all the models compared, the improved CEEMDAN-FE-TCN model obtains the lowest MAE, RMSE, GEH average at 2.10, 2.96, and 0.39, respectively, and the highest R-square at 0.968. Though the error of all models increases with the prediction step prolonging, the proposed model performs best on two-step and three-step ahead predictions, indicating its goodness of fit on long- and short-term predictions.

TCN, LSTM, and GRU all realize the memory of the long-term changing features. Therefore, they appear to be more accurate with the extensive training samples by extracting deeper traffic dynamics than the SVR and HA models, reducing the prediction error. However, unlike the RNN models, TCN can capture the whole long-term sequence features by convolving parallelly. So, it takes up less memory and avoids forgetting information, which thoroughly learns the global time series characteristics and achieves more accuracy than RNN. Furthermore, under the framework of the improved CEEMDAN-FE-X, the TCN RNN and SVR models all perform better than the direct prediction, which means the decomposition

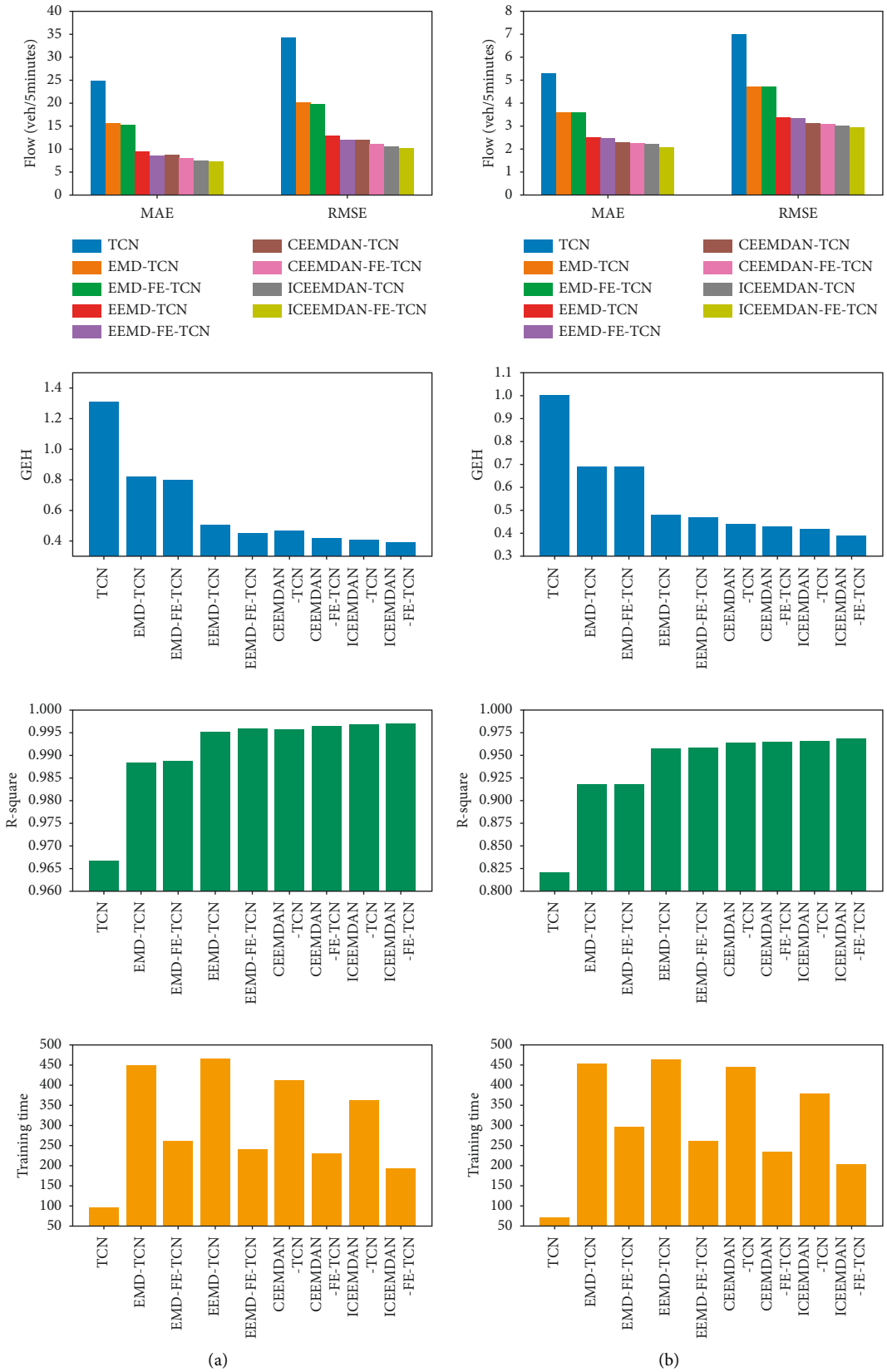


Figure 10: Prediction performance of different models based on TCN: (a) 717490 and (b) 718462.

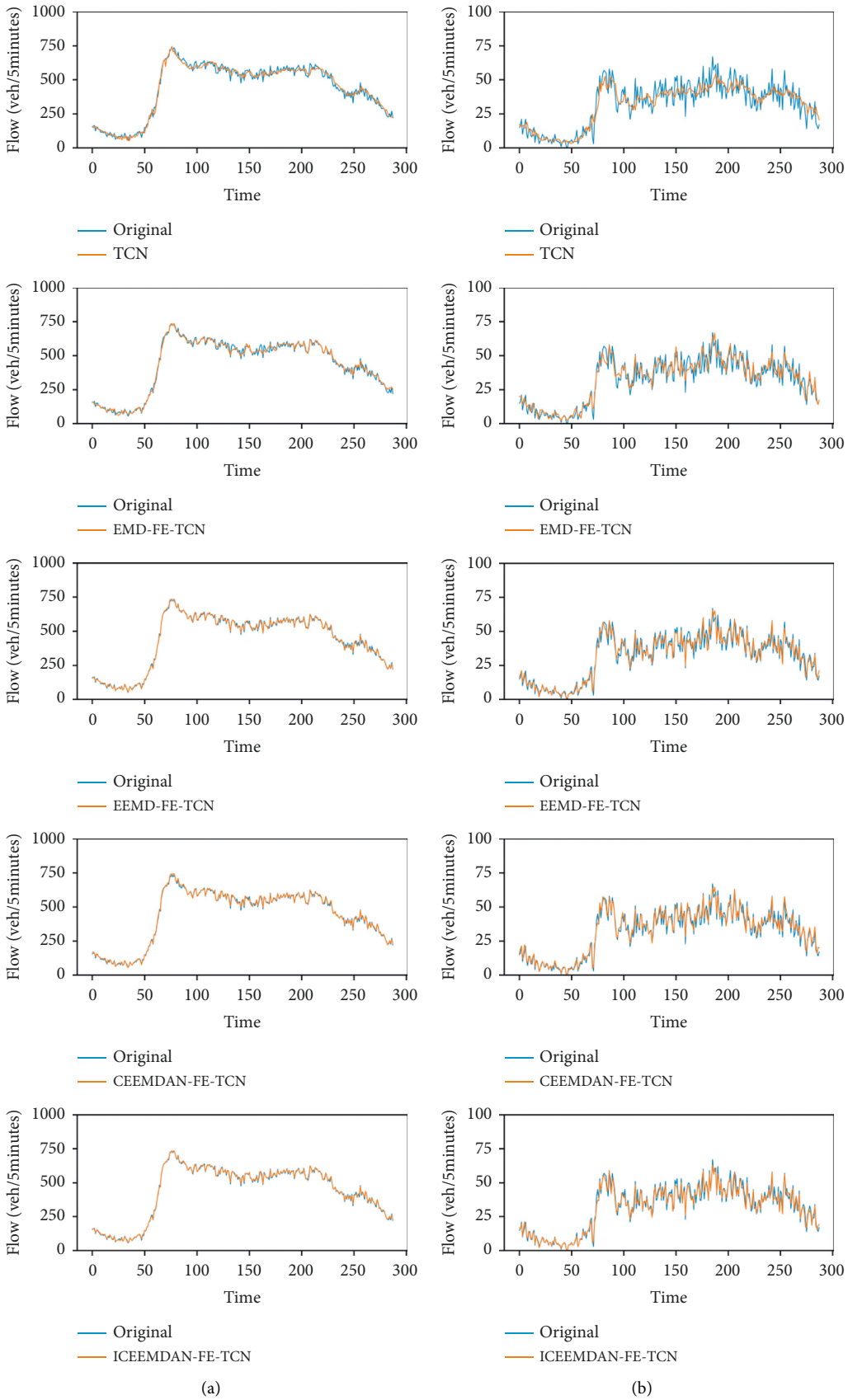


Figure 11: Prediction performance comparison on Aug 31, 2018, of different models based on TCN: (a) 717490 and (b) 718462.

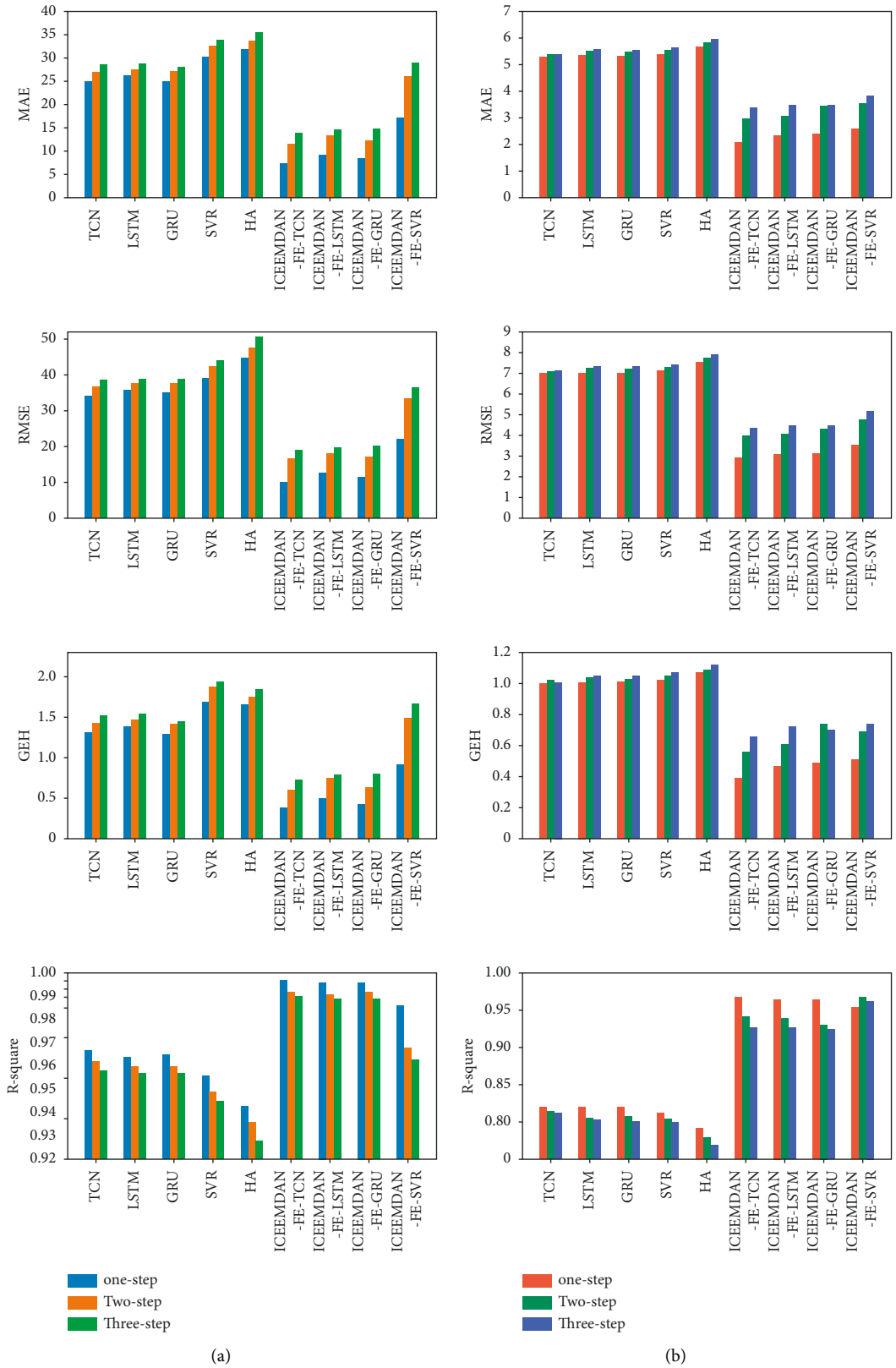


Figure 12: Prediction performance of different algorithms: (a) 717490 and (b) 718462.

Table 7: Prediction error of different algorithms of 717490 and 718462.

| 717490 | | | | | 718462 | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Evaluation criterion | One-step | Two-step | Three-step | Model | Evaluation criterion | One-step | Two-step | Three-step |
| TCN | MAE | 24.98 | 27.00 | 28.59 | TCN | MAE | 5.31 | 5.41 | 5.41 |
| | RMSE | 34.33 | 36.79 | 38.75 | | RMSE | 7.01 | 7.11 | 7.16 |
| | GEH (Average) | 1.31 | 1.43 | 1.52 | | GEH (Average) | 1.00 | 1.02 | 1.01 |
| | R-square | 0.967 | 0.962 | 0.958 | | R-square | 0.820 | 0.815 | 0.812 |
| LSTM | MAE | 26.22 | 27.56 | 28.80 | LSTM | MAE | 5.35 | 5.52 | 5.58 |
| | RMSE | 35.83 | 37.74 | 38.91 | | RMSE | 7.02 | 7.28 | 7.34 |
| | GEH (Average) | 1.39 | 1.47 | 1.54 | | GEH (Average) | 1.01 | 1.04 | 1.05 |
| | R-square | 0.964 | 0.960 | 0.957 | | R-square | 0.820 | 0.806 | 0.803 |
| GRU | MAE | 25.06 | 27.23 | 28.10 | GRU | MAE | 5.34 | 5.50 | 5.58 |
| | RMSE | 35.03 | 37.88 | 38.99 | | RMSE | 7.02 | 7.24 | 7.37 |
| | GEH (Average) | 1.30 | 1.42 | 1.45 | | GEH (Average) | 1.01 | 1.03 | 1.05 |
| | R-square | 0.965 | 0.960 | 0.957 | | R-square | 0.820 | 0.808 | 0.801 |
| SVR | MAE | 30.30 | 32.69 | 33.80 | SVR | MAE | 5.41 | 5.55 | 5.64 |
| | RMSE | 39.33 | 42.53 | 44.15 | | RMSE | 7.17 | 7.31 | 7.42 |
| | GEH (Average) | 1.69 | 1.87 | 1.94 | | GEH (Average) | 1.02 | 1.05 | 1.07 |
| | R-square | 0.956 | 0.949 | 0.945 | | R-square | 0.812 | 0.804 | 0.799 |
| HA | MAE | 31.92 | 33.70 | 35.55 | HA | MAE | 5.69 | 5.83 | 5.96 |
| | RMSE | 44.89 | 47.70 | 50.66 | | RMSE | 7.55 | 7.76 | 7.95 |
| | GEH (Average) | 1.66 | 1.75 | 1.85 | | GEH (Average) | 1.07 | 1.09 | 1.12 |
| | R-square | 0.943 | 0.936 | 0.928 | | R-square | 0.792 | 0.780 | 0.769 |
| ICEEMDAN-FE-TCN | MAE | 7.36* | 11.57* | 13.87* | ICEEMDAN-FE-TCN | MAE | 2.10* | 2.97* | 3.39* |
| | RMSE | 10.34* | 16.93* | 19.09* | | RMSE | 2.96* | 3.99* | 4.36* |
| | GEH (Average) | 0.39* | 0.60* | 0.73* | | GEH (Average) | 0.39* | 0.56* | 0.66* |
| | R-square | 0.997* | 0.992* | 0.990* | | R-square | 0.968* | 0.942* | 0.927* |
| ICEEMDAN-FE-LSTM | MAE | 9.32 | 13.37 | 14.66 | ICEEMDAN-FE-LSTM | MAE | 2.34 | 3.09 | 3.49 |
| | RMSE | 12.61 | 18.25 | 19.71 | | RMSE | 3.10 | 4.08 | 4.48 |
| | GEH (Average) | 0.50 | 0.74 | 0.79 | | GEH (Average) | 0.47 | 0.61 | 0.72 |
| | R-square | 0.996 | 0.991 | 0.989 | | R-square | 0.965 | 0.939 | 0.927 |
| ICEEMDAN-FE-GRU | MAE | 8.41 | 12.32 | 14.91 | ICEEMDAN-FE-GRU | MAE | 2.39 | 3.45 | 3.49 |
| | RMSE | 11.68 | 17.26 | 20.12 | | RMSE | 3.14 | 4.33 | 4.51 |
| | GEH (Average) | 0.43 | 0.64 | 0.80 | | GEH (Average) | 0.49 | 0.74 | 0.70 |
| | R-square | 0.996 | 0.992 | 0.989 | | R-square | 0.964 | 0.931 | 0.925 |
| ICEEMDAN-FE-SVR | MAE | 17.13 | 26.08 | 28.95 | ICEEMDAN-FE-SVR | MAE | 2.60 | 3.54 | 3.85 |
| | RMSE | 22.31 | 33.57 | 36.40 | | RMSE | 3.55 | 4.79 | 5.16 |
| | GEH (Average) | 0.92 | 1.49 | 1.67 | | GEH (Average) | 0.51 | 0.69 | 0.74 |
| | R-square | 0.986 | 0.968 | 0.963 | | R-square | 0.954 | 0.916 | 0.903 |

*Note.* *indicates the best results; ICEEMDAN means improved CEEMDAN.

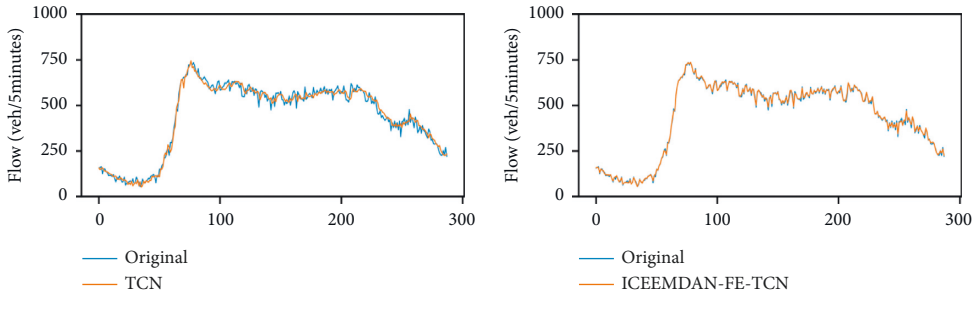


Figure 13: Continued.

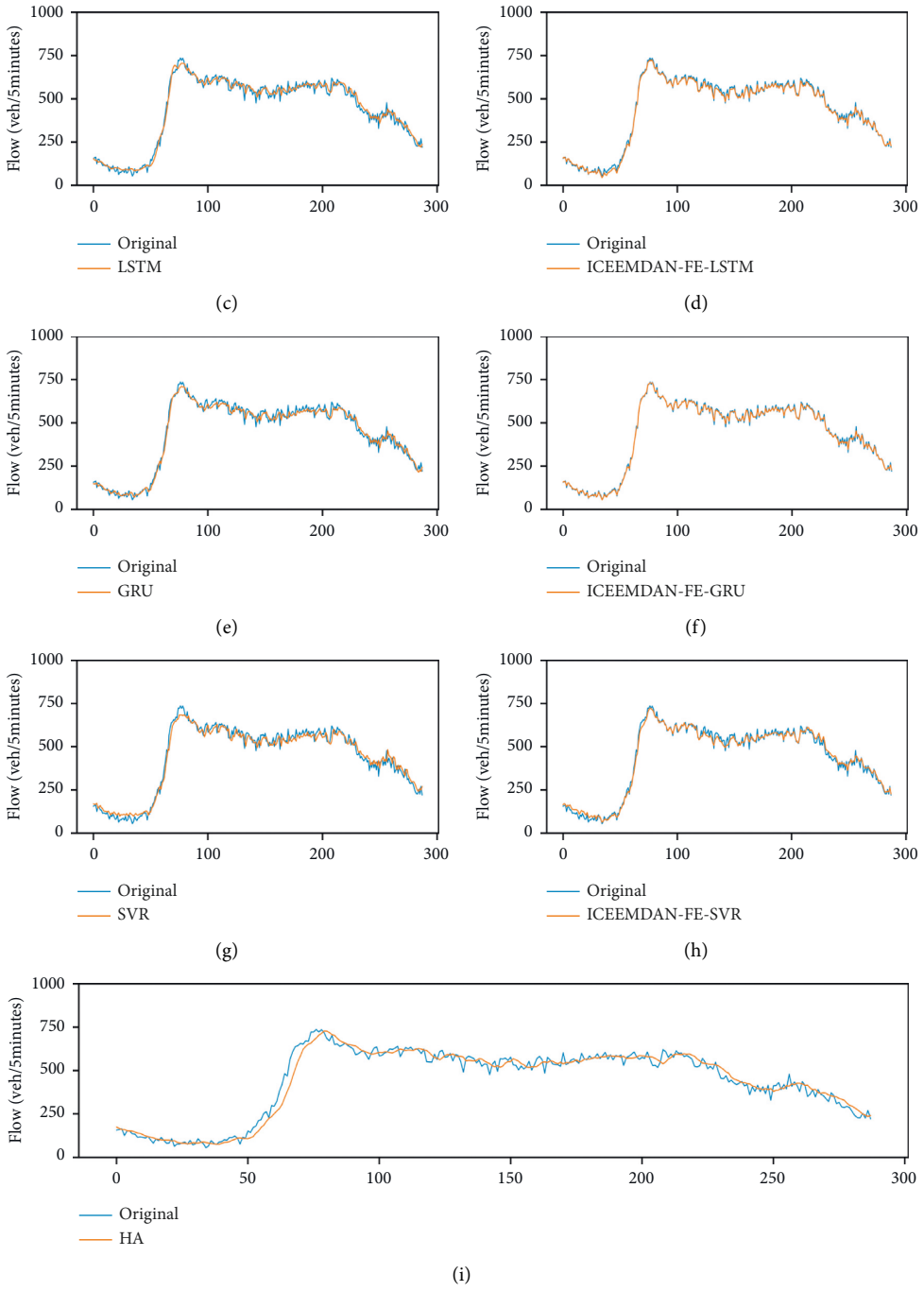


Figure 13: One-step ahead prediction performance comparison on Aug 31, 2018, of different algorithms of 717490.

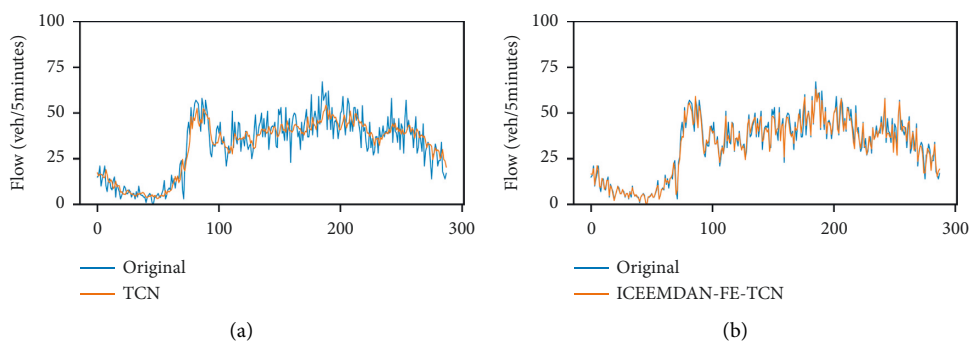


Figure 14: Continued.

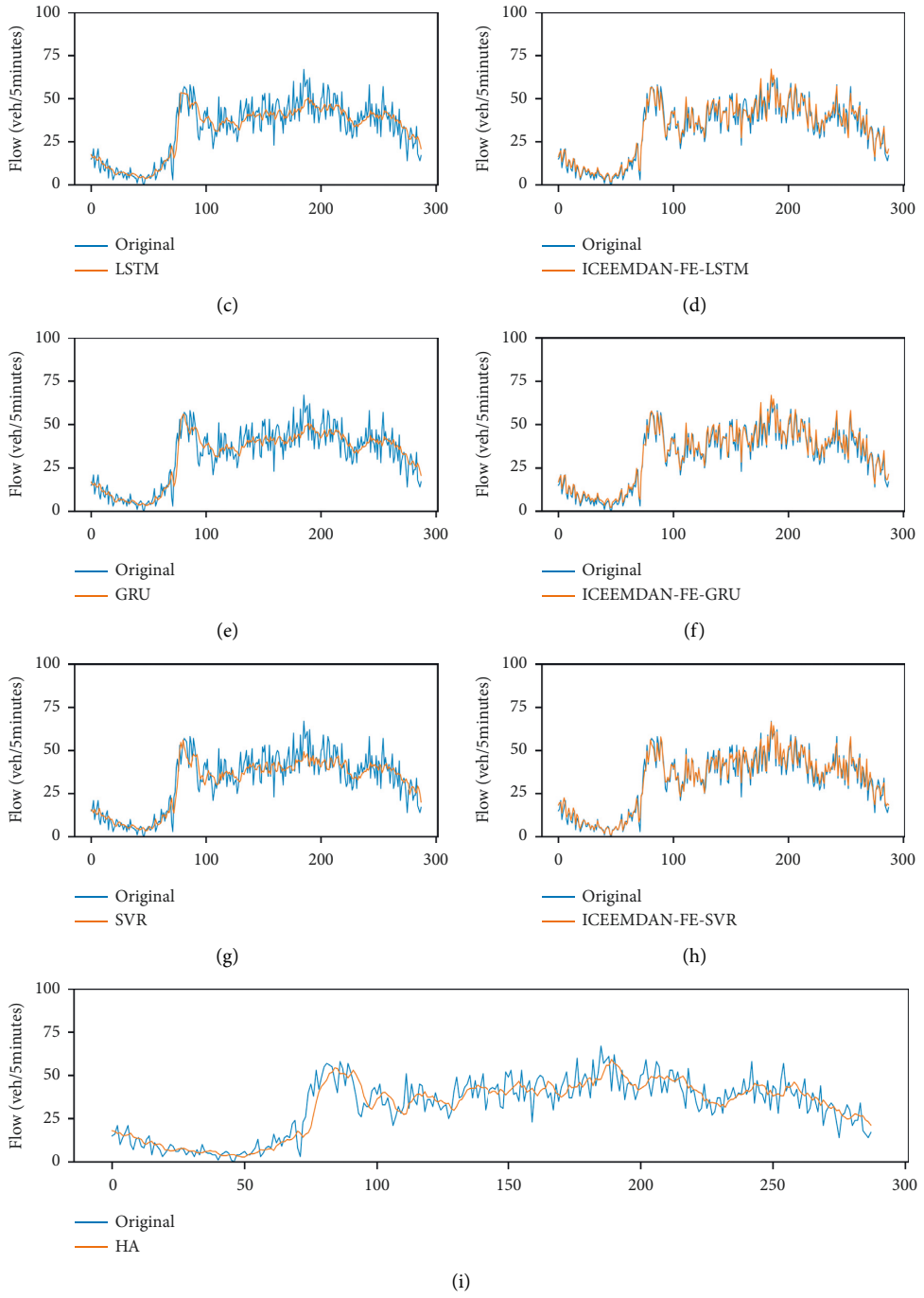


Figure 14: One-step ahead prediction performance comparison on Aug 31, 2018, of different algorithms of 718462.

prediction has the universality on different prediction models. What should be mentioned is that the proposed framework has the best optimization effect on the TCN model and has a better effect on the RNN model than on the SVR model for both the mainline and the ramp flow. In conclusion, under the reasons mentioned above, the improved CEEMDAN-FE-TCN outperforms the other models compared in this paper on the highway mainline and ramp traffic flow prediction.

## 7. Conclusions

In this paper, an improved CEEMDAN-FE-TCN model is proposed to forecast highway traffic flow. First, the improved CEEMDAN method decomposes the nonlinear highway traffic flow into IMF and residual with different frequencies. Then, the FE of each mode is calculated, and the modes with similar chaos are recombined as subsequences, highlighting the traffic dynamics. Finally, the TCN is applied to predict

the different recombined subsequences. After reconstructing the output of TCN submodels, the predicted traffic flow data is obtained. The data of two sensors on US101-S: VDS No. 717490 and VDS No. 718462 collected from PeMS were tested. Compared with other models, the following conclusions are drawn:

(1) The improved CEEMDAN algorithm can decompose the traffic flow time series with different frequencies. The accuracy of time series prediction after decomposition and reconstruction is notably higher than direct prediction. Compared with conventional EMD-based models, the improved CEEMDAN-FE-TCN obtains the lowest prediction error.

(2) The FE algorithm can calculate the chaos of the modes decomposed by the original data. By recombining the modes with similar FE, the main dynamics of traffic flow are highlighted while amplifying the details of fluctuations. The prediction efficiency and accuracy would be further improved.

(3) The improved CEEMDAN-FE-X framework has remarkable effects on single-step and multistep traffic flow prediction. Under this structure, the prediction accuracy of the TCN, LSTM, GRU, and SVR models is significantly increased. The proposed model outperforms the other models in this paper on the highway mainline and ramp traffic flow prediction, confirming the robustness.

Studies can be combined with other aspects in future work, such as adding spatial factors into the time series prediction. Decomposing the spatiotemporal graph and selecting suitable models for the subgraphs of different frequencies to make predictions may improve accuracy.

## Data Availability

All of the data related to this paper are available on Caltrans PeMS (https://pems.dot.ca.gov/).

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation Item (Grant no. 52072143).

## References

[1] C. Ma, J. Zhou, X. (Daniel) Xu, and J. Xu, "Evolution regularity mining and gating control method of urban recurrent traffic congestion: a literature review," *Journal of Advanced Transportation*, vol. 2020, Article ID 5261580, 13 pages, 2020.

[2] S. Suhas, V. V. Kalyan, M. Katti, A. B. V. Prakash, and C. Naveena, "A comprehensive review on traffic prediction for intelligent transport system," in *Proceedings of the 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*, pp. 138–143, Bangalore, India, March 2017.

[3] L. N. N. Do, N. Taherifar, and H. L. Vu, "Survey of neural network-based models for short-term traffic state prediction," *Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery*, vol. 9, no. 1, Article ID e1285, 2019.

[4] M. Akhtar and S. Moridpour, "A review of traffic congestion prediction using artificial intelligence," *Journal of Advanced Transportation*, vol. 2021, Article ID 8878011, 18 pages, 2021.

[5] J. Li, F. Guo, A. Sivakumar, Y. Dong, and R. Krishnan, "Transferability improvement in short-term traffic prediction using stacked LSTM network," *Transportation Research Part C: Emerging Technologies*, vol. 124, Article ID 102977, 2021.

[6] F. Alesiani, L. Moreira-Matias, and M. Faizrahnemoon, "On learning from inaccurate and incomplete traffic flow data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3698–3708, 2018.

[7] M. Ramezani and E. Ye, "Lane density optimisation of automated vehicles for highway congestion control," *Transportmetrica B-Transport Dynamics*, vol. 7, no. 1, pp. 1096–1116, 2019.

[8] W. Zhang, G. Han, X. Wang, M. Guizani, K. Fan, and L. Shu, "A node location algorithm based on node movement prediction in underwater acoustic sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3166–3178, 2020.

[9] W. Li, J. Wang, R. Fan et al., "Short-term traffic state prediction from latent structures: accuracy vs. efficiency," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 72–90, 2020.

[10] L. N. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, and D. Phung, "An effective spatial-temporal attention based neural network for traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 108, pp. 12–28, 2019.

[11] T. Ma, C. Antoniou, and T. Toledo, "Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast," *Transportation Research Part C-Emerging Technologies*, vol. 111, pp. 352–372, 2020.

[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] Y. Qi and S. Ishak, "A hidden Markov model for short term prediction of traffic conditions on freeways," *Transportation Research Part C-Emerging Technologies*, vol. 43, pp. 95–111, 2014.

[14] C. H. Wu, J. M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[15] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, 2016.

[16] A. Csikos, Z. J. Viharos, K. B. Kis, T. Tettamanti, and I. Varga, "Traffic speed prediction method for urban networks-an ANN approach," in *Proceedings of the 2015 International Conference on Models and Technologies for Intelligent Transportation*

*Systems (MT-ITS)*, pp. 102–108, Budapest, Hungary, June 2015.

[17] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[18] H. Nguyen, L. M. Kieu, T. Wen, and C. Cai, "Deep learning methods in transportation domain: a review," *IET Intelligent Transport Systems*, vol. 12, no. 9, pp. 998–1004, 2018.

[19] H. Zhang, Z. Wang, and D. Liu, "A comprehensive review of stability analysis of continuous-time recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 7, pp. 1229–1262, 2014.

[20] W. Guoli, "Traffic prediction and attack detection approach based on PSO optimized Elman neural network," in *Proceedings of the 2019 11th International Conference on Measuring Technology and Mechatronics Automation*, pp. 504–508, Qiqihar, China, April 2019.

[21] S. Yang, K. Xie, and P. Shi, "A TDNN based urban short-term traffic follow prediction model," *Dynamics of Continuous Discrete and Impulsive Systems: Series B; Applications and Algorithms*, vol. 14, pp. 695–700, 2007.

[22] X. Zhao and M. Yang, "Using NARX neural network for prediction of urban rail transit passenger flow," in *Proceedings of the 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 117–121, Beijing, China, November 2018.

[23] 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," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[24] Y. Gao, J. Zhao, Z. Qin, Y. Feng, Z. Yang, and B. Jia, "Traffic speed forecast in adjacent region between highway and urban expressway: based on MFD and GRU model," *Journal of Advanced Transportation*, vol. 2020, Article ID 8897325, 18 pages, 2020.

[25] J. Tang, J. Zeng, Y. Wang, H. Yuan, F. Liu, and H. Huang, "Traffic flow prediction on urban road network based on license plate recognition data: combining attention-LSTM with genetic algorithm," *Transportmetrica: Transportation Science*, vol. 17, no. 4, pp. 1217–1243, 2021.

[26] P. Wu, Z. Huang, Y. Pian, L. Xu, J. Li, and K. Chen, "A combined deep learning method with attention-based LSTM model for short-term traffic speed forecasting," *Journal of Advanced Transportation*, vol. 2020, Article ID 8863724, 15 pages, 2020.

[27] T. Li, A. Ni, C. Zhang, G. Xiao, and L. Gao, "Short-term traffic congestion prediction with Conv–BiLSTM considering spatio-temporal features," *IET Intelligent Transport Systems*, vol. 14, no. 14, pp. 1978–1986, 2020.

[28] R. L. Abduljabbar, H. Dia, and P. W. Tsai, "Unidirectional and bidirectional LSTM models for short-term traffic prediction," *Journal of Advanced Transportation*, vol. 2021, Article ID 5589075, 15 pages, 2021.

[29] J. Gao, X. Gao, and H. Yang, "Short-term traffic flow prediction based on time-Space characteristics," in *Proceedings of the 2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE)*, pp. 128–132, Beijing, China, September 2020.

[30] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, https://arxiv.org/abs/1803.01271.

[31] W. Zhao, Y. Gao, T. Ji, X. Wan, F. Ye, and G. Bai, "Deep temporal convolutional networks for short-term traffic flow forecasting," *IEEE Access*, vol. 7, Article ID 114507, 2019.

[32] R. Zhang, F. Sun, Z. Song, X. Wang, Y. Du, and S. Dong, "Short-term traffic flow forecasting model based on GA-TCN," *Journal of Advanced Transportation*, vol. 2021, Article ID e1338607, 2021.

[33] N. E. Huang, Z. Shen, S. R. Long et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society A-Mathematical Physical and Engineering Sciences*, vol. 454, pp. 679–699, 1998.

[34] N. E. Huang, M. L. C. Wu, S. R. Long et al., "A confidence limit for the empirical mode decomposition and Hilbert spectral analysis," *Proceedings of the Royal Society A-Mathematical Physical and Engineering Sciences*, vol. 459, no. 2037, pp. 2317–2345, 2003.

[35] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method," *Advances in Adaptive Data Analysis*, vol. 1, no. 1, pp. 1–41, 2009.

[36] J. R. Yeh, J. S. Shieh, and N. E. Huang, "Complementary ensemble empirical mode decomposition: a novel noise enhanced data analysis method," *Advances in Adaptive Data Analysis*, vol. 2, no. 2, pp. 135–156, 2010.

[37] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4144–4147, Prague, Czech Republic, May 2011.

[38] M. A. Colominas, G. Schlotthauer, and M. E. Torres, "Improved complete ensemble EMD: a suitable tool for biomedical signal processing," *Biomedical Signal Processing and Control*, vol. 14, pp. 19–29, 2014.

[39] Y. Wei and M. C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 148–162, 2012.

[40] X. Chen, H. Chen, Y. Yang et al., "Traffic flow prediction by an ensemble framework with data denoising and deep learning model," *Physica A: Statistical Mechanics and its Applications*, vol. 565, Article ID 125574, 2021.

[41] 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, Article ID 42054, 2020.

[42] Z. Wang, R. Chu, M. Zhang, X. Wang, and S. Luan, "An improved hybrid highway traffic flow prediction model based on machine learning," *Sustainability*, vol. 12, p. 20, 2020.

[43] 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, Article ID 51004, 2021.

[44] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1003–1012, LA, USA, July 2017.

[45] H. Zhao, M. Sun, W. Deng, and X. Yang, "A new feature extraction method based on EEMD and multi-scale fuzzy entropy for motor bearing," *Entropy*, vol. 19, p. 14, 2016.

[46] W. Chen, Z. Wang, H. Xie, and W. Yu, "Characterization of surface EMG signal based on fuzzy entropy," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 2, pp. 266–272, 2007.

[47] W. Chen, J. Zhuang, W. Yu, and Z. Wang, "Measuring complexity using FuzzyEn, ApEn, and SampEn," *Medical Engineering & Physics*, vol. 31, no. 1, pp. 61–68, 2009.

[48] A. V. D. Oord, S. Dieleman, H. Zen et al., "WaveNet: a generative model for raw audio," 2016, https://arxiv.org/abs/1609.03499.