Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5257353, 16 pages
https://doi.org/10.1155/2022/5257353

Research Article

Interval Short-Term Traffic Flow Prediction Method Based on CEEMDAN-SE Nosie Reduction and LSTM Optimized by GWO

Wenzheng Zhao,[1] **Yuer Yang**,[2] **and Zihao Lu**[3]

[1]*College of Engineering, China Agricultural University, Beijing 100000, China*
[2]*College of Cyber Security, Jinan University; School of Economics, Jinan University, Guangzhou 510632, China*
[3]*School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, Heilongjiang 150080, China*

Correspondence should be addressed to Wenzheng Zhao; zhaowenzheng1999@gmail.com

Received 10 May 2022; Revised 26 June 2022; Accepted 1 July 2022; Published 10 August 2022

Academic Editor: Kuruva Lakshmanna

With rapid economic growth and urbanization, the accelerated increase in car ownership has brought massive pressure on urban traffic, and accurate traffic flow prediction information can provide an important basis for urban traffic dynamic planning. The existing methods have problems such as low efficiency, large error, and inability to adapt to short-term traffic changes. To solve the above problems, the CEEMDAN-SE-GWO-LSTM method was proposed in this paper. First, the traffic flow data is processed for outliers and missing values. The Complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) method is used to decompose the traffic flow data, and sample entropy (SE) is used to reconstruct the subsequence, which is used to improve the quality of the input data. Then, the Grey Wolf Optimizer (GWO) is used to optimize the parameters of the long-short-term memory (LSTM) in order to improve the prediction accuracy and prevent the model from falling into a local optimum. Three models are used to compare with the ensemble model proposed in this paper, including back propagation neural network (BPNN), LSTM, and long-short-term memory optimized by Grey Wolf Optimizer (GWO-LSTM). Root mean square error (RMSE) is reduced by 40.9% to 66.7%; R^2 score is improved by 1.5% to 7.1%. The experimental results show that CEEMDAN-SE-GWO-LSTM has a higher prediction accuracy than the existing traffic flow prediction models. Finally, this paper uses the model prediction error to establish an interval prediction model based on the kernel density estimation theory, which enhances the generalization of the model and the practical application value.

1. Introduction

1.1. Background. Traffic congestion has become increasingly severe, generating social problems such as prolonged travel times and frequent traffic accidents. These issues can be effectively alleviated by collecting and processing traffic flow data and building an intelligent transportation system (ITS) [1]. The recent works on intelligent transportation system are summarized in Table 1.

Traffic flow forecasting, as an important part of the traffic system, can effectively reduce traffic congestion. Furthermore, accurate traffic prediction information can efficiently help local governments allocate traffic resources to reduce traffic congestion. Moreover, the forecast results can then aid travelers in planning their traffic routes and thus reduce travel time.

1.2. Literature Review. In cities, various factors can influence traffic flow, such as weather, geography, and time of day. These factors are predominately highly nonlinear and volatile. Therefore, if the collected data is used directly in the experiment, the model will not accurately discover the changing pattern of traffic flow time series, resulting in low prediction accuracy. To further reduce the effect of noise on model predictions, Chen and Chou [8] proposed an empirical mode decomposition (EMD) method, which decomposes traffic flow signals to intrinsic mode functions (IMFs) by using EMD theory. Ensemble empirical mode

Table 1: Recent works on intelligent transportation system.

Number	Solved problem	Model name
1	Sensor data is analyzed using crow search algorithm optimized long-short-term memory to correctly identify drivers [2].	Crow search algorithm optimized long-short-term memory (CSA-LSTM)
2	Spatio-temporal individual mobility graph encoding network with group mobility assistance (SIGMA) is proposed to encode individual mobility behavior, which enables recommendation of new locations [3].	Spatio-temporal individual mobility graph encoding network with group mobility assistance (SIGMA)
3	This work proposes a deep learning-based traffic safety solution in 5G intelligent transportation systems that can effectively predict drivers' intention to change lanes [4].	(1) Lane-change intention recognition based on an LSTM and historical driving-track data (2) Lane-change intention recognition based on an LSTM and natural-driving data (3) Recognition of lane-change intention based on decision layer fusion
4	This work presents an edge node deep learning-based traffic flow detection scheme that combines vehicle detection and vehicle tracking algorithms and is deployed to the edge device Jetson TX2 platform [5].	A vehicle detection network based on improved YOLOv3 and a vehicle tracking network based on the improved DeepSORT
5	This work proposes a dynamic and intelligent traffic light control system (DITLCS) that dynamically adjusts traffic light durations by analyzing real-time traffic information, which improves the efficiency of traffic light control systems [6].	Deep reinforcement learning and fuzzy inference system
6	This work presents a radial basis function neural network algorithm based on quantum particle swarm optimization (QPSO) strategy for traffic flow prediction in intelligent transportation system (ITS) [7].	Quantum particle swarm optimization (QPSO) strategy

decomposition (EEMD) improves the mode mixing of EMD by adding Gaussian white noise to the original sequence. Liu et al. used EEMD to decompose the traffic flow data [9]. CEEMDAN improves the processing of EEMD and achieves better decomposition results with higher computational efficiency. Lu et al. used the CEEMDAN method to decompose the raw traffic flow [10]. This paper uses the SE to reorganize the IMFs obtained from CEEMDAN to complete the dimension reduction processing of traffic flow data.

Traffic flow can be predicted due to its strong regularity and periodicity, but its uncertainty increases the difficulty of prediction [11]. Many efforts have been made to forecast STTF, and these efforts can be broadly classified into parametric and nonparametric methods. Autoregressive integrated moving average (ARIMA) is a standard parametric method for forecasting time series data. Yu and Zhang proposed switching the ARIMA model and applied it to actual data obtained from UTC/SCOOT system [12]. Kumar and Vanajakshi proposed a seasonal ARIMA (SARIMA) model for STTF prediction [13]. Chen et al. proposed an autoregressive integrated moving average with generalized autoregressive conditional heteroscedasticity (ARIMA-GARCH) model to predict traffic flow [14].

Another standard parametric method for time series prediction is the Kalman filter technique. Kumar proposed a Kalman filter technique (KFT) model for traffic flow prediction [15]. Guo et al. proposed an adaptive Kalman filter approach to predict STTF prediction [16]. However, parametric methods are limited by assumptions such as smoothness of the time series, which may lead to poor accuracy when the time series varies irregularly. Therefore, the parametric approach has limited applicability in the transportation field. Unlike parametric methods, which are limited by several preconditions, nonparametric methods have become the mainstream strategy for STTF prediction. Yang and Lu proposed a combined wavelet-SVM prediction model for STTF prediction [17]. Duan used a particle swarm optimization (PSO) algorithm to select the appropriate learning parameters of support vector machine (SVM) for STTF prediction [18]. Alam et al. applied five regression models to predict the traffic flow in the city of Porto [19]. However, the predictions calculated by these methods are still unsatisfactory and need to be more precise in their application. With the development of deep learning and data volume growth, traffic flow prediction methods based on deep learning have been gradually proposed and achieved good results.

Zhang et al. proposed a convolutional neural network- (CNN-) based deep learning framework for STTF prediction [20]. Zheng and Huang used a long-short-term memory (LSTM) network to predict traffic flow data [1]. Qu et al. proposed a new end-to-end improved LSTM model, M-B-LSTM, to predict STTF [21]. Ma et al. used a convolutional neural network (CNN) to extract traffic flow pattern features, and the extracted features were fed into an LSTM unit [22]. Zhao et al. proposed a temporal convolutional network (TCN) model to predict STTF in the city [23]. LSTM is widely used to predict time series among the algorithms related to neural networks. However, this algorithm has significant complexities and has the disadvantage of not achieving global optimality. This paper uses the Grey Wolf Optimizer (GWO) to optimize the parameters of the LSTM network, which avoids the situation that the LSTM algorithm falls into local optimum. In this paper, to reduce the effect of variable nonstationarity on the prediction, we decomposed the natural traffic flow data using the CEEMDAN-SE method. Then, we used the Grey Wolf Optimizer (GWO) LSTM algorithm for traffic flow prediction.

1.3. Contribution and Paper Framework. The main contributions of the paper are as follows.

(i) Noise reduction of the data is performed by the CEEMDAN-SE method. Due to the volatility and instability of traffic flow, using raw data directly as the input to the model can lead to low prediction accuracy. In this paper, we use CEEMDAN to decompose the original sequences and use SE to measure the complexity of each IMF component. Moreover, we combined the subsequences with similar complexities to reduce input dimension and improve prediction efficiency

(ii) An optimized GWO-LSTM model is built to predict the traffic flow data. Since the traditional LSTM model is prone to fall into local optimum, this paper adopts the GWO algorithm to optimize the parameters of the LSTM, which improves the model's optimization-seeking speed and prediction accuracy

(iii) Multimodel comparison experiments are constructed. The CEEMDAN-SE-GWO-LSTM model proposed in this paper is compared with BPNN, LSTM, and GWO-LSTM. The SSE, MAE, MSE, MAPE, RMSE, and R^2 metrics are calculated separately for the above models to quantify prediction accuracy

(iv) Use the kernel density estimation function to estimate the model prediction error distribution and establish an interval prediction model

The rest of the paper is organized as follows. In Section 2, the paper describes the signal decomposition algorithm used, the optimization algorithm, the deep learning model, and the probabilistic interval estimation method. In Section 3, the paper presents the experiments performed, which mainly include the results of the decomposition of the traffic flow, the tuning of the hyperparameters by the optimization algorithm, the comparison of the prediction results of the proposed model with other models, and the results of the interval prediction. In Section 4, the work done in this paper is summarized, and the limitations of this research as well as the future directions of development are described.

2. Proposed Method

This chapter mainly introduces our proposed STTF prediction model. It is well known that traffic flow is a signal that varies nonstationarily over time (frequency varies over time). Previously, there were no excellent theories for nonstationary processing signals. The EMD decomposition [24] proposed by Huang et al. in 1998, part of the Hilbert-Huang transform (HHT), was a breakthrough in this kind of signal analysis. Huang et al.'s method was based upon spectral decomposition. Spectral decomposition decomposes the signal into components of different frequencies. In our proposal, we adopted the theory of CEEMDAN [25] (an improved version of EMD) to decompose the original nonstationary traffic flow signal. We decomposed the signals into several IMF subsignals of different frequencies as the input of the prediction model GWO-LSTM to improve the model's prediction accuracy.

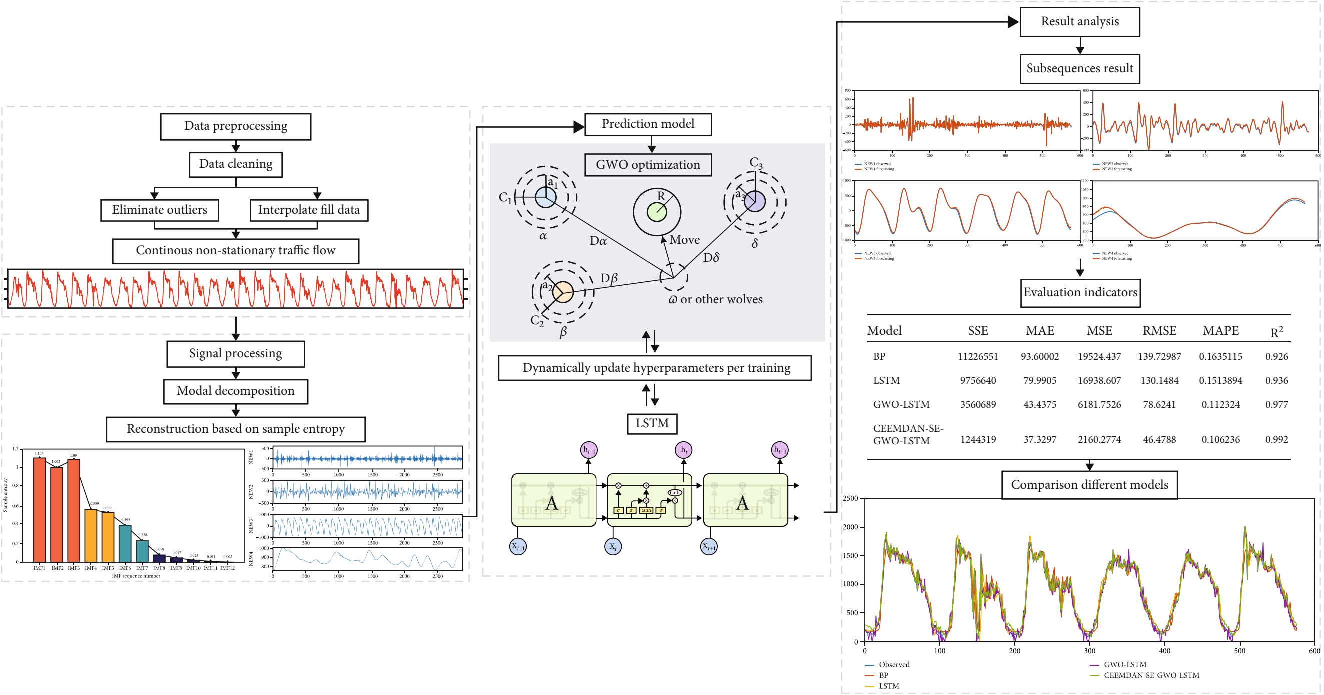
Furthermore, this paper utilised SE to measure the nonlinear complexity of the IMF subsequence processed by CEEMDAN [26] to reduce the dimension of the IMF subsequence. Thus, SE was introduced to solve the problem of excessive computational scale. In the prediction model, we utilised the LSTM model based on GWO optimisation to improve the prediction accuracy and reduce the training time by optimising parameters such as the number of nodes, iterations, and learning rate. The superiority of our proposed CEEMDAN-SE-GWO-LSTM is demonstrated by comparison with other benchmark models. We will introduce our model in detail through the following three aspects: decomposition of nonstationary traffic flow signals, SE-based IMF subsignal fusion, and STTF prediction based on GWO-LSTM. This work's general arrangement is shown in Figure 1.

2.1. CEEMDAN-SE

2.1.1. CEEMDAN. EMD (empirical mode decomposition) is a classic adaptive method for solving nonstationary signal problems [18]. However, the modal aliasing problem of EMD will cause severe sawtooth lines in the time-frequency distribution. It makes certain eigenmode functions lose their physical meaning, which leads to the degradation of the performance of EMD.

Based on the CEEMD-SE method, Wang et al. [27] presented a wind power short-term prediction model. In their experiment, the RMSE and MAE were 2.16 and 0.39, respectively, better than EMD-SE-HS-KELM, HS-KELM, KELM, and ELM models. In 2020, Tian [28] presented a STTF prediction model. The model was based upon EMD method and combination model; their proposed model demonstrated superior performance in STTF prediction. However, the two models have limitations. Neither model takes noise into account; neither can automatically adjust to follow changes in noise.

Therefore, we propose utilizing CEEMDAN [25] method to decompose the traffic flow time series to reduce its nonstationarity. CEEMDAN is an EMD-based algorithm, and it uses the EEMD method to add Gaussian noise to the original signal. Then, the signal decomposition performance is improved. The mode mixing problem is solved by the idea of multiple stacking and averaging operation to cancel the influence of noise to obtain better mode decomposition results. The process of the CEEMDAN algorithm is shown in Figure 2.



Model	SSE	MAE	MSE	RMSE	MAPE	R^2
BP	11226551	93.60002	19524.437	139.72987	0.1635115	0.926
LSTM	9756640	79.9905	16938.607	130.1484	0.1513894	0.936
GWO-LSTM	3560689	43.4375	6181.7526	78.6241	0.112324	0.977
CEEMDAN-SE-GWO-LSTM	1244319	37.3297	2160.2774	46.4788	0.106236	0.992

Figure 1: The framework of this paper.

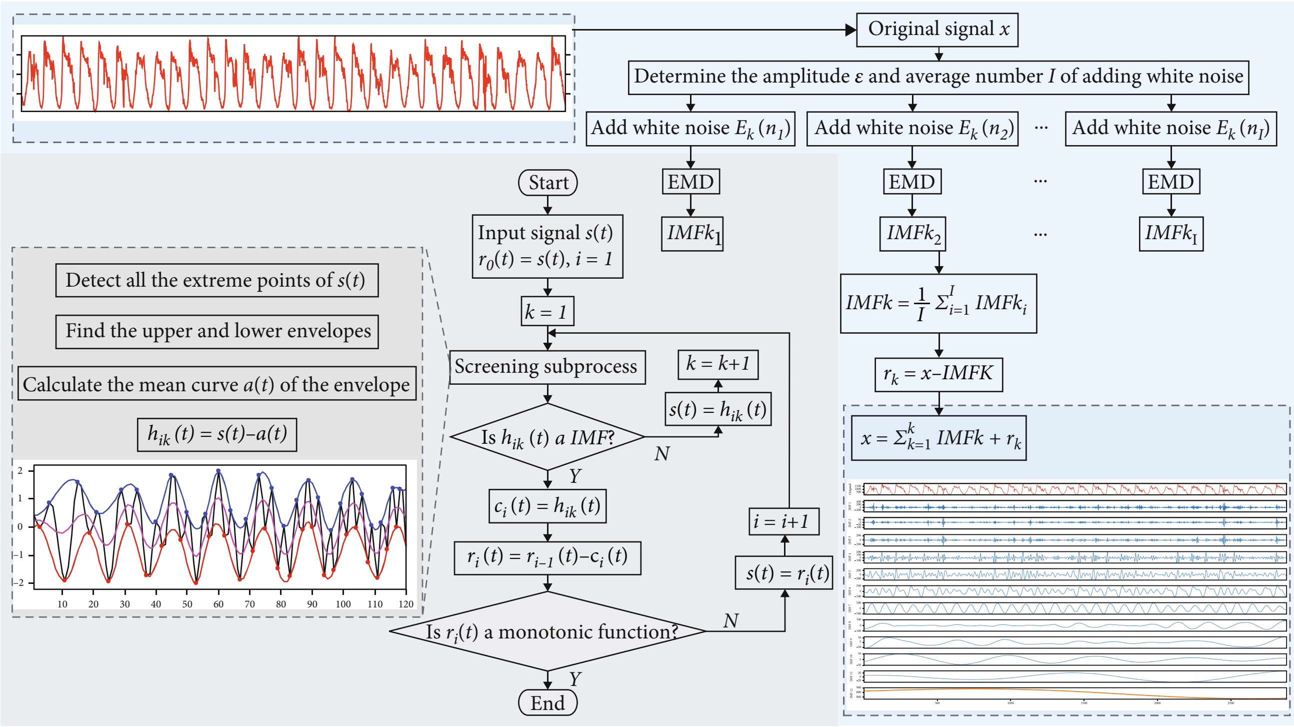

Original signal x
Determine the amplitude ε and average number I of adding white noise
Add white noise $E_k(n_1)$
Add white noise $E_k(n_2)$
···
Add white noise $E_k(n_I)$
EMD
$IMFk_1$
$IMFk_2$
$IMFk_I$
$IMFk = \frac{1}{I}\sum_{i=1}^{I} IMFk_i$
$r_k = x - IMFK$
$x = \sum_{k=1}^{k} IMFk + r_k$
Start
Input signal $s(t)$
$r_0(t) = s(t),\ i = 1$
$k = 1$
Screening subprocess
$k = k+1$
$s(t) = h_{ik}(t)$
Is $h_{ik}(t)$ a IMF?
N
Y
$c_i(t) = h_{ik}(t)$
$r_i(t) = r_{i-1}(t) - c_i(t)$
$i = i+1$
$s(t) = r_i(t)$
Is $r_i(t)$ a monotonic function?
End
Detect all the extreme points of $s(t)$
Find the upper and lower envelopes
Calculate the mean curve $a(t)$ of the envelope
$h_{ik}(t) = s(t) - a(t)$

Figure 2: The process of the CEEMDAN algorithm.

2.1.2. Sample Entropy Theory. N subsequences will be generated after the CEEMDAN decomposition of the traffic flow-time series data. Using them directly as the input data of the GWO-LSTM model will result in a sizeable computational scale. Therefore, SE, a nonlinear complexity measure, is used to classify and reconstruct the traffic flow-time series samples to reduce the complexity of subsequences. SE [29] is a method based on approximate entropy (ApEn) [26], which evaluates time series complexities by measuring the probability of generating new patterns in time series signals. SE is defined as the negative natural logarithm of the conditional probability, where self-matches are not included.

$$\mathrm{SampEn}(m, r) = \lim_{N \longrightarrow \infty} \left\{ -\ln \left[\frac{A^m(r)}{B^m(r)} \right] \right\}. \tag{1}$$

$B^m(r)$ in Equation (1) is the probability of the two sequences matching m points under the similarity tolerance r, and $A^m(r)$ is the probability of the two sequences matching $m + 1$ points. The calculation formulas are Equation (4) and Equation (5), respectively.

$$A_i^m(r) = \frac{1}{N - m - 1} A_i, \tag{2}$$

$$B_i^m(r) = \frac{1}{N - m - 1} B_i, \tag{3}$$

$$A^m(r) = \frac{1}{N - m} \sum_{i=1}^{N-m} A_i^m(r), \tag{4}$$

$$B^m(r) = \frac{1}{N - m} \sum_{i=1}^{N-m} B_i^m(r). \tag{5}$$

A_i and B_i are the number of the maximum distance, not greater than r, between the vector sequences $X_m(i)$ and $X_m(j)$ of the dimension m composed of time series data when the dimension is $m + 1$ and m, respectively. Specifically, $X_m(i) = \{x(i), x(i + 1), \cdots, x(i + m - 1)\}$, $1 \le i \le N - m + 1$, represents m consecutive values of x starting from the ith point.

The amount of data is usually limited in specific applications. Thus, Equation (1) evolved into

$$\mathrm{SampEn}(m, r, N) = - \ln \left[\frac{A^m(r)}{B^m(r)} \right]. \tag{6}$$

2.2. GWO-LSTM. LSTM was first presented in 1997 [30] as an algorithm that could make machines learn much faster and help solve complex artificial long-time-lag tasks. As the GWO algorithm [31] imitates the hunting mechanism that the leadership hierarchy of grey wolves in nature, it could be used to optimize the LSTM model. Our presented model has a significant optimization effect compared to LSTM neural networks and BP neural networks. Thus, this paper uses GWO-LSTM method to complete the postprocessing of CEEMDAN-SE. The general process of the GWO-LSTM framework is shown in Figure 3.

2.2.1. LSTM. LSTM is designed to have the problem in long-long-term dependence solved. Compared to RNN, LSTM has three more gates—forgetting gate, input gate, and output gate—enabling it to achieve better results in traffic flow prediction.

Since the output is a linear combination of the inputs, we need to enhance the nonlinearity of LSTM. The enhancement will be done through the use of the activation function as it exacerbates the nonlinearity of the network model. Common activation functions for LSTM are tanh (-1, 1), sigmoid (0, 1) and relu [0, 1). Following experimental verification, tanh (-1, 1) presents better results to our problem and is selected as our activation function.

(1) GWO. We first divided the traffic flow prediction into four layers and entered them into the GWO model to complete the initialization, with the first three layers being of greater significance. We defined α as the optimum solution. During the hunt, the behavior of grey wolves rounding up their prey was defined as Equation (7) and Equation (8), where t is the current iterative generation, $\boldsymbol{A}$ and $\boldsymbol{C}$ are the coefficient vectors, and $\boldsymbol{X}_p$ and $\boldsymbol{X}$ are the prey position vector and the grey wolf position vector, respectively.

$$\boldsymbol{D} = \left| \boldsymbol{C}\boldsymbol{X}_p(t) - \boldsymbol{X}(t) \right|, \tag{7}$$

$$\boldsymbol{X}(t + 1) = \boldsymbol{X}_p(t) - \boldsymbol{A}\boldsymbol{D}. \tag{8}$$

The calculation equations of $\boldsymbol{A}$ and $\boldsymbol{C}$ are shown in Equation (9) and Equation (10), where α is the convergence factor. As iterations decreases linearly from 2 to 0, the norms of r_1 and r_2 are random numbers between [0, 1].

$$\boldsymbol{A} = 2\alpha r_1 - \alpha, \tag{9}$$

$$\boldsymbol{C} = 2r_2. \tag{10}$$

In the GWO model, the upper layer leads the lower layer to the set of update equations shown in Equation (11), and after completing the update, the GWO model outputs $\boldsymbol{X}(t + 1)$ to the LSTM model according to Equation (12). Subsequently, the model calculates the loss function and adjusts the learning rate of the GWO model according to the vector $\boldsymbol{X}$.

$$\begin{cases} D_\alpha = \left| C_1 \boldsymbol{X}_\alpha - \boldsymbol{X} \right|, \\ D_\beta = \left| C_2 \boldsymbol{X}_\beta - \boldsymbol{X} \right|, \\ D_\delta = \left| C_3 \boldsymbol{X}_\delta - \boldsymbol{X} \right|, \end{cases} \tag{11}$$

$$X(t + 1) = \frac{(X_\alpha - A_1 D_\alpha) + \left(X_\beta - A_2 D_\beta\right) + (X_\delta - A_3 D_\delta)}{3}. \tag{12}$$

(2) GWO-LSTM. We referenced the data on the LSTM model to derive a prediction of the baseline model. Subsequently, we incorporated the LSTM prediction results into the GWO model to obtain the new four strata. Once the four strata are obtained, GWO will calculate the coefficient matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ according to Equation (9) and Equation (10). Then, it will calculate the ratios of each stratum in the four strata using **A** and **C**, inputting them into the LSTM model for automated parameter tuning, and continue to train the

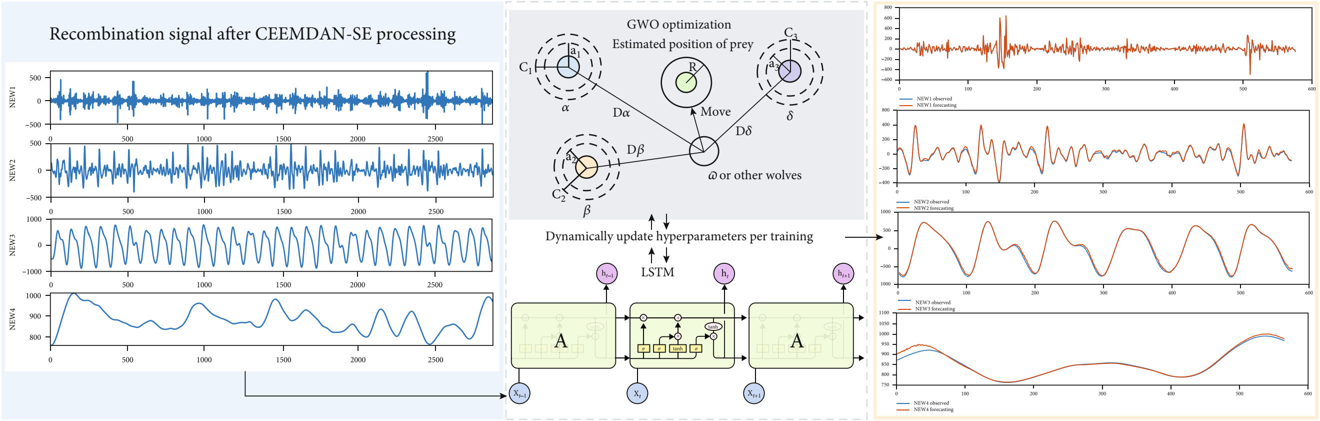


Figure 3: The presented GWO-LSTM framework.

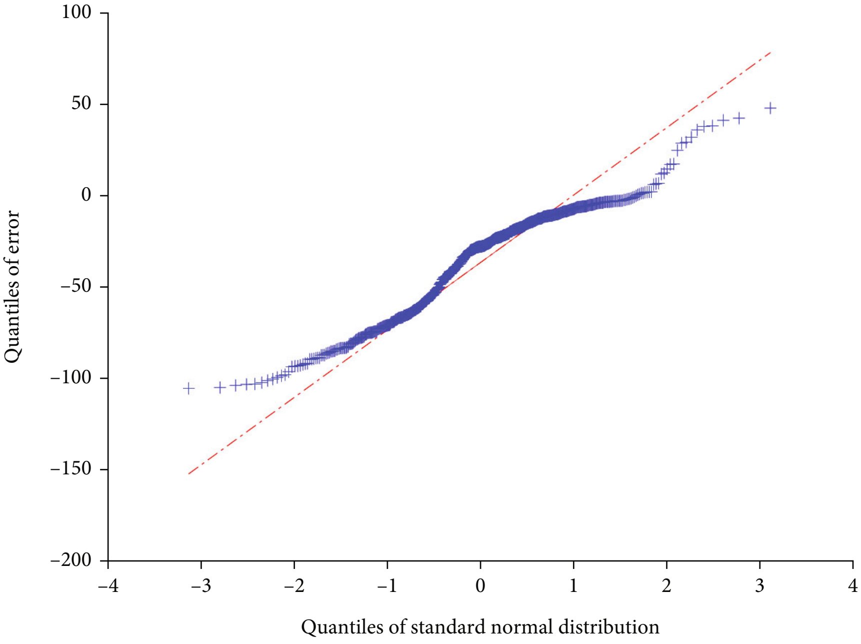


Figure 4: *Q-Q* diagram of short-term traffic flow prediction error.

LSTM model. The above process will then repeat until a user-specified number of iterations is reached.

Machine learning training aims to update the parameters and optimize the objective function. In this paper, the GWO is set as an optimizer to perform a local estimation ($|\mathbf{A}|$ shown as Equation (9)) based on the results of each LSTM iteration to minimize the loss function. We use 1024 as the initial batch size and 0.01 as the initial learning rate. GWO decides to update or not update (eliminate or not eliminate) the population of grey wolves based on the results of each LSTM iteration and, thus, dynamically adjusts the learning rate of the LSTM each time. In addition, our GWO network is optimized for four layers.

Finally, this paper constructs a CEEMDAN-SE-GWO-LSTM model combining GWO-LSTM with CEEMDAM-SE to obtain a more accurate traffic flow prediction. Repeating the above process, when $|\mathbf{A}|$ is less than 1, the local optimum is trapped at this time; when $|\mathbf{A}|$ is greater than 1, the global optimum is reached.

2.3. Probability Interval Prediction. We define the STTF prediction error ε as the deviation between the actual observed value P_{obs} of traffic flow and the predicted value P_{pred} of traffic flow at a certain moment, defined as

$$\varepsilon = P_{\text{obs}} - P_{\text{pred}}. \tag{13}$$

Then, we use the historical data of prediction error to perform probability density function fitting; the purpose is to avoid the contingency of STTF error accuracy using kernel density estimation (KDE). Finally, we use the inverse cumulative distribution function (ICDF) to expand the prediction results into intervals to improve the prediction accuracy and generalization.

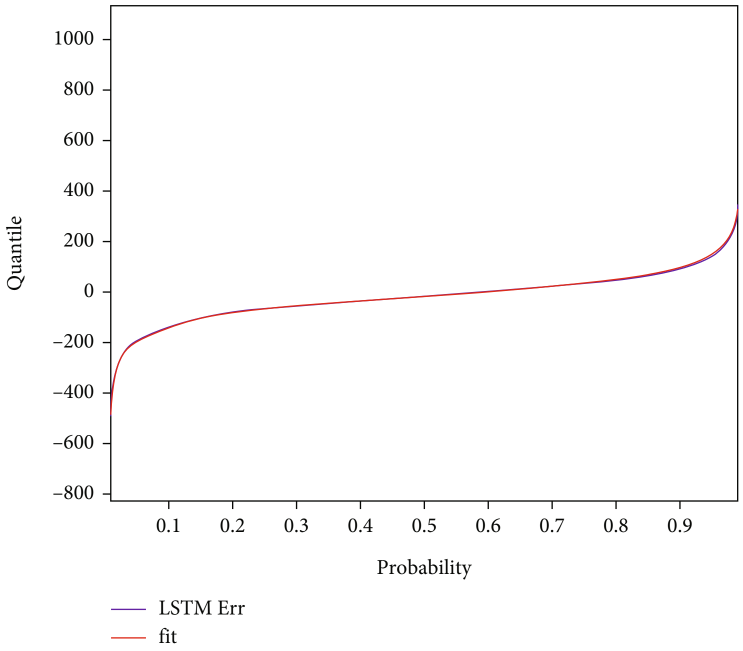


Figure 5: ICDF of the prediction error of LSTM.

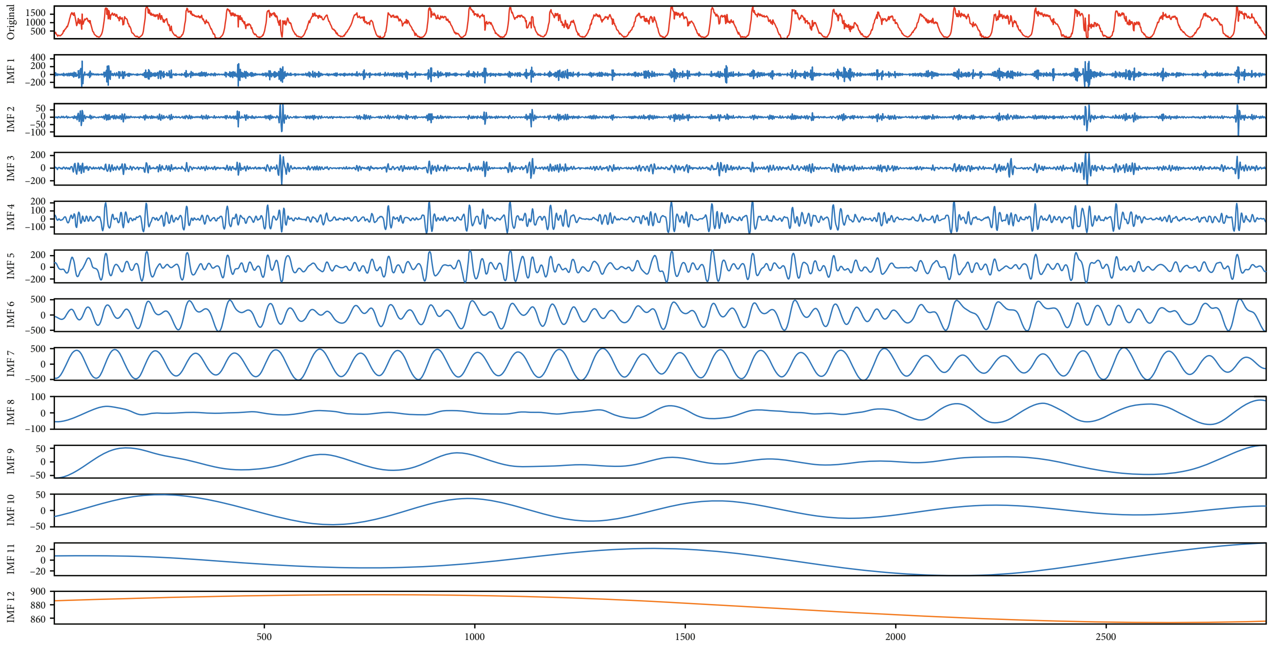


Figure 6: The decomposition results of CEEMDAN.

Due to the randomness of STTF and its dependence on time characteristics, the prediction error of STTF does not conform to the assumption of normal distribution, as is shown in Figure 4 that the error distribution of our proposed method does not belong to the normal distribution after being discriminated by the Q-Q diagram.

Therefore, the normal distribution based on the parameter estimation method is not suitable for the confidence estimation of STTF problems. Correspondingly, nonparametric estimation methods do not require prior assumptions about the distribution of prediction errors. Instead, it fits a probability distribution according to the input data, which is more adaptable to nonstationary data.

As a nonparametric probability density estimation method, KDE has a stable fitting effect, so this paper uses KDE to predict the confidence interval of traffic flow. KDE

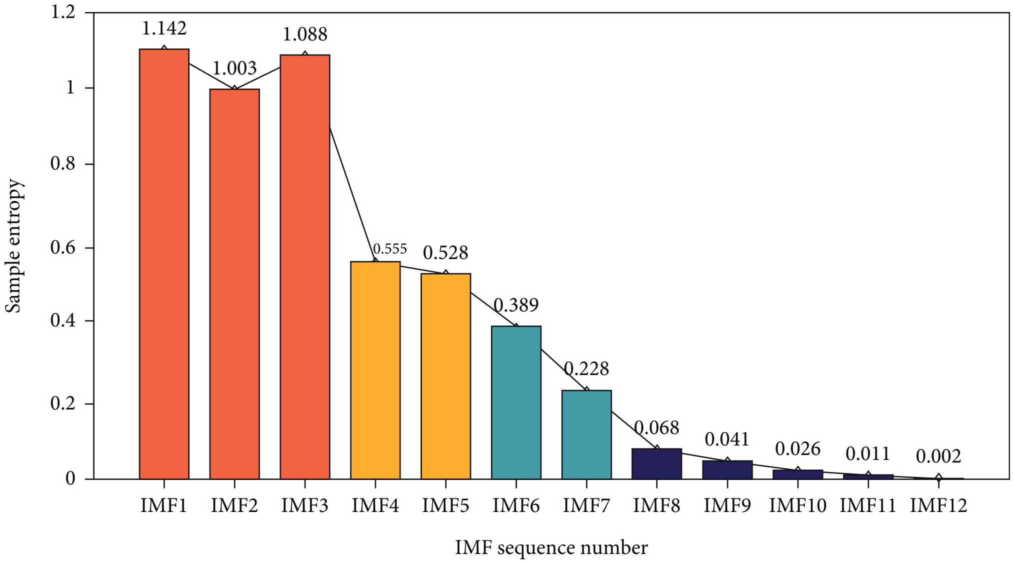


Figure 7: The sample entropy of each subsequence.

Table 2: The reconstructed subordinate list.

IMFn	Hse (n)	Merged sequence	New IMF component
IMF1	1.142	IMF1, IMF2, IMF3	NEW1
IMF2	1.003		
IMF3	1.088		
IMF4	0.555	IMF4, IMF5	NEW2
IMF5	0.528		
IMF6	0.389	IMF6, IMF7	NEW3
IMF7	0.228		
IMF8	0.068	IMF8, IMF9, IMF10, IMF11, IMF12	NEW4
IMF9	0.041		
IMF10	0.026		
IMF11	0.011		
IMF12	0.002		

assigns the probability of each point to nearby intervals, and the final density function can be obtained by accumulating the interval densities of all points.

Assuming the sample set $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}$ of traffic sequence data, all sample points obey the distribution $f(x)$, let the function $\widehat{f}(x)$ be the KDE of $f(x)$, $\widehat{f}(x)$ can be expressed as

$$\widehat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{14}$$

In the upon formula, x_i represents the ith sample in the dataset, h is the window width, which represents the interval division size of the sample error distribution, and $K(x)$ is the kernel function, which determines the role of each sample point x_i in density estimation. In practical applications, the Gaussian kernel function is the most widely used due to its good mathematical properties. This paper selects the Gaussian kernel function for kernel density estimation, and its expression is as

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}. \tag{15}$$

2.4. Inverse Cumulative Distribution Function. The inverse cumulative distribution function gives the value associated with a specific cumulative probability. Therefore, this paper uses ICDF to determine the confidence interval. That is, we want to expand the prediction results into an interval and discuss whether the method can include observations in our prediction interval, thereby further improving the prediction accuracy [32].

To get the ICDF, the cumulative distribution function (CDF) of the error function needs to be calculated first. The cumulative distribution function of a random variable X is a function on the real numbers that is denoted as F and is given by Equation (16), where x is for any $x \in$ R.

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt. \tag{16}$$

The ICDF and CDF are inverse to each other, so we draw the ICDF diagram under different algorithms, as shown in Figure 5, which is the ICDF diagram of the error distribution of the LSTM algorithm on the dataset used in this paper. The ICDF and CDF are inverse to each other, so we draw the ICDF diagram under different algorithms, as shown in the figure, the ICDF diagram of the error distribution of the LSTM algorithm on the dataset used in this paper. Then, we take the 80%, 90%, and 95% confidence intervals, respectively, and calculate their error intervals under different

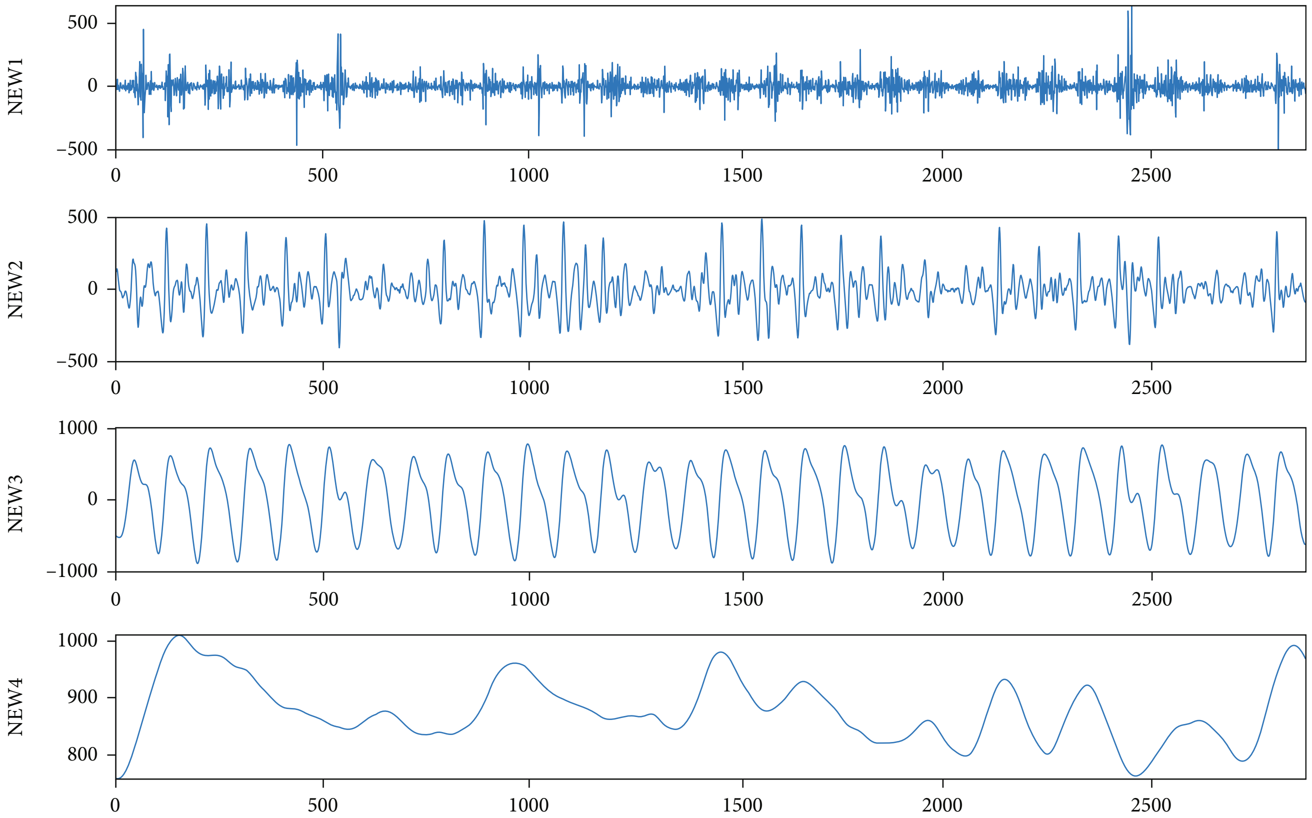


Figure 8: The reconstitution components of heat load through CEEMDAN-SE.

Table 3: The prediction results of GWO-LSTM.

Features	Iterations	LSTM		GWO-LSTM	
		Train	Test	Train	Test
1/2	100	0.0319	0.0291	0.0210	0.0197
1/2	200	0.0305	0.0282	0.0153	0.0125
1/2	300	0.0269	0.0253	0.0109	0.0068
3/4	100	0.0286	0.0253	0.0219	0.0205
3/4	200	0.0260	0.0244	0.0175	0.0186
3/4	300	0.0245	0.0226	0.0130	0.0152
2/3/4	100	0.0218	0.0188	0.0090	0.0107
2/3/4	200	0.0185	0.0169	0.0081	0.0069
2/3/4	300	0.0160	0.0120	0.0075	0.0048

algorithms, so the predicted value can be expanded based on the error interval. Thereby, a prediction result with bandwidth is generated. It makes the prediction model results more accurate, and the specific results will be shown in the experiments chapter.

3. Experiment

The section in question will thoroughly examine the related experiments. Notably, the CEEMDAN-SE and the GWO-LSTM were initially tested on our datasets. Finally, we combined the two models and created CEEMDAN-SE-GWO-LSTM. CEEMDAN-SE-GWO-LSTM will then be placed into direct comparison with other state-of-the-art models to evidence how it surpasses other models.

3.1. Sample Selection and Data Sources. Among the most significant ways of travel, flying remains one of the most prevalent means in which the public opts. The geographical location of an airport generally is distant from an individual's area of residence. Hence, providing accurate traffic information is essential for planning in advance. This paper utilizes the measured traffic flow data of high-speed stations near the M25 Heathrow Airport to ensure the reliability and authenticity of predicated results. The dataset is the traffic flow data for 30 consecutive days from September 1, 2019, to September 30, 2019, with a collection frequency of 15 minutes. The data volume became 2880 ensuing the interpolation method. The interpolation method was applied to fill in the missing values of the time series and remove the outliers. In this paper, we uniformly divided the dataset into the training set and test set according to the ratio of 8 : 2.

3.2. CEEMDAN-SE. We first employed the CEEMDAN method to decompose the traffic flow. After the CEEMDAN decomposition (as shown in Figure 6), we obtained the first 11 IMF components with different complexities and one IMF residual component with a relatively gentle change.

In order to avoid the input signal data being too large, we introduced the theory of SE. We introduced said theory in order to group and reconstruct the IMF subsignals that were obtained after decomposition. Figure 7 is the SE of the IMF

Table 4: The result of different models.

Model	SSE	MAE	MSE	RMSE	MAPE	R^2
BP	11226551	93.60002	19524.4368	139.72987	0.16351147	0.926
LSTM	9756640	79.9905	16938.607	130.1484	0.1513894	0.936
GWO-LSTM	3560689	43.4375	6181.7526	78.6241	0.112324	0.977
CEEMDAN-SE-GWO-LSTM	**1244319**	**37.3297**	**2160.2774**	**46.4788**	**0.106236**	**0.992**

component. The IMF subfunctions of similar SE were superimposed to reduce the input sample dimension, then feed it into the GWO-LSTM model.

In this paper, the entropy value of each component (as shown in Figure 7) is used as the judging standard; each IMF component is reorganized. Specifically, IMF1, IMF2, and IMF3 have similar complexities and thus can be combined into recombined components. Similarly, the remaining IMF components are merged and recombined; the recombination result is shown in Table 2, and the new subsequence after reconstruction is shown in Figure 8.

3.3. GWO-LSTM. Experiments are conducted based on the above data to verify the effectiveness of the GWO-LSTM model on traffic flow prediction. The initialized batch size is 1024, and the learning rate is 0.01. The number of implied layers of the LSTM network chosen are two. The learning rate of the LSTM model, thus, is continuously and dynamically adjusted in the iterative process using the GWO algorithm with four implied layers until the loss function is obtained below a predefined specific value. In the experiments of this paper, we uniformly set the number of iterations to 500 to train GWO-LSTM and CEEMDAN-SE-GWO-LSTM.

In order to obtain better results from the CEEMDAN-SE-GWO-LSTM model, we need to pretrain the GWO-LSTM. Our optimization target is the loss function; the loss values of LSTM and GWO-LSTM with different feature combinations and iteration numbers are shown in Table 3. The numbers in column features shown in Table 3 stand for the corresponding layers that the GWO optimizes.

3.4. Comparative Analysis of Prediction Models. To further illustrate the effectiveness of the CEEMDAN-SE-GWO-LSTM model, the BP model, standard LSTM model, and the improved GWO-LSTM model was selected for comparison.

As more error evaluation indexes are found in traffic flow prediction, six commonly indexes are used to evaluate the model in this paper. The related equations of the six error evaluation indexes mentioned are shown from Equation (17) to Equation (22), where n is the number of test set data, and y_i and $\hat{y}_i$ are the actual traffic flow value and predicted traffic flow value at moment i, respectively.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\%, \tag{17}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \tag{18}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2, \tag{19}$$

$$\mathrm{SSE} = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2, \tag{20}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|, \tag{21}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}. \tag{22}$$

The results of different models are shown in Table 4, and the comparison of prediction results of different models is shown in Figures 9 and 10.

Table 4 shows the quantitative results of different models, and it can be seen that our method performs the best among all the compared models. From a numerical point of view, the proposed model is 88.9%, 60.1%, 88.9%, 66.7%, and 35% lower in the SSE, MAE, MSE, RMSE, and MAPE evaluation indicators of the BP comparison model, and 7.1% higher in the R^2 evaluation indicator. Similarly, the values of our model for LSTM model are 87.2%, 53.3%, 87.2%, 64.2%, 29.8%, and 6%, respectively, and the values for GWO-LSTM are 65.1%, 14.1%, 65.1%, 40.9%, 5.4%, and 1.5%, respectively.

For the analysis of different models, Figure 10(a) is the result of the BP neural network. The limitation of the BP network is that it has a significant error in the prediction of extreme values, and traffic congestion often occurs in the period when the extreme values are generated. Figure 10(b) is the result of the LSTM model. The predicted value can better reflect the traffic flow trend, and it performs well at extreme values. However, when the traffic flow has a long-term trend, it has a hysteresis, which cannot perfectly solve the problem of timely forecasting. Figure 10(c) is GWO-LSTM, and it shows that the error performs well in each quantization index. However, the forecast results will have large fluctuations at certain times; this is due to the characteristics of LSTM and nonstationary signals. When LSTM uses the node information of adjacent nonstationary signals, it will be affected by a considerable fluctuation value and generate a significant prediction error. Figures 9 and 10(d) show the prediction results of each subsequence and the sum of each subsequence, respectively. Our proposed modal decomposition can better suppress this noise effect.

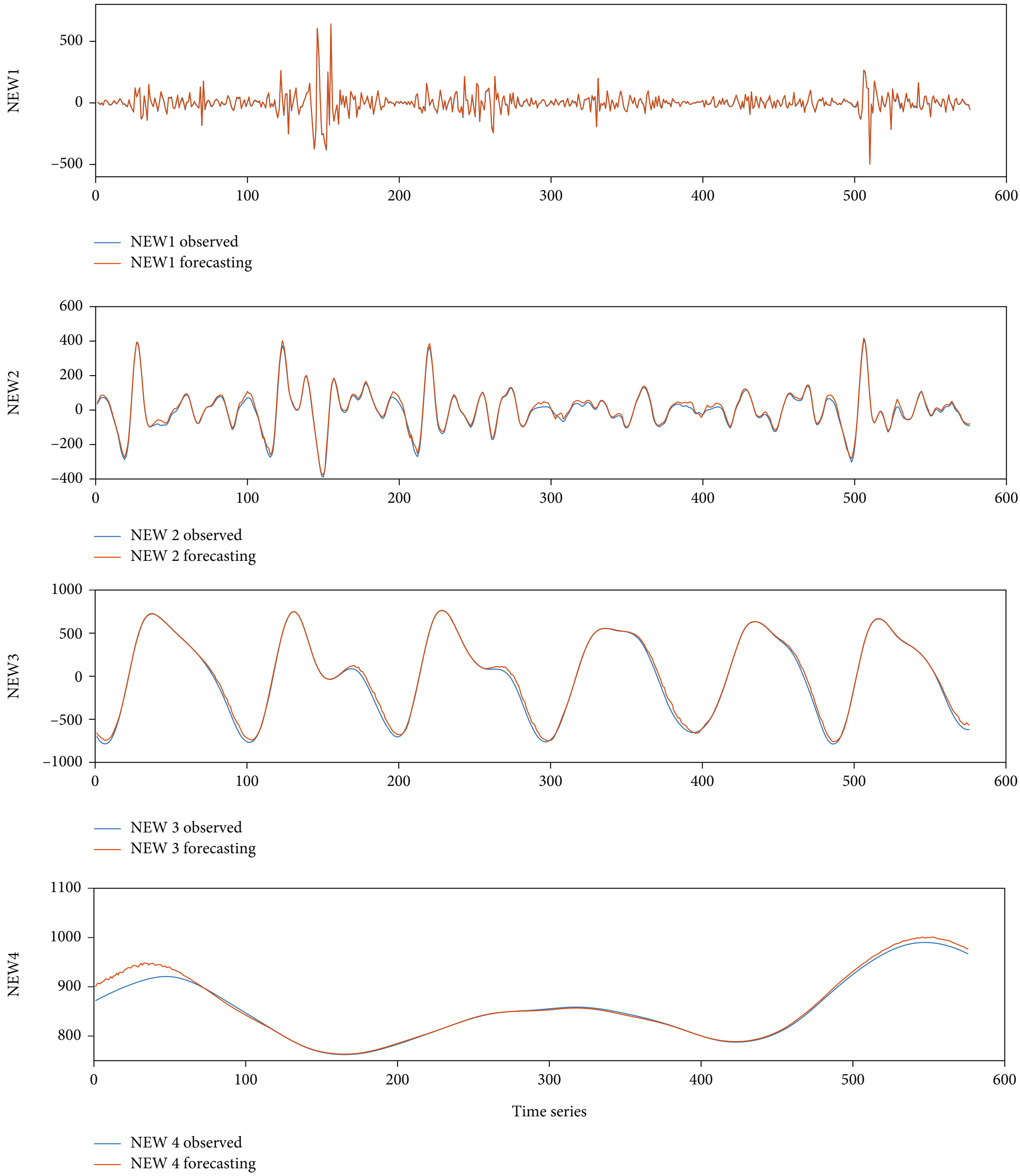
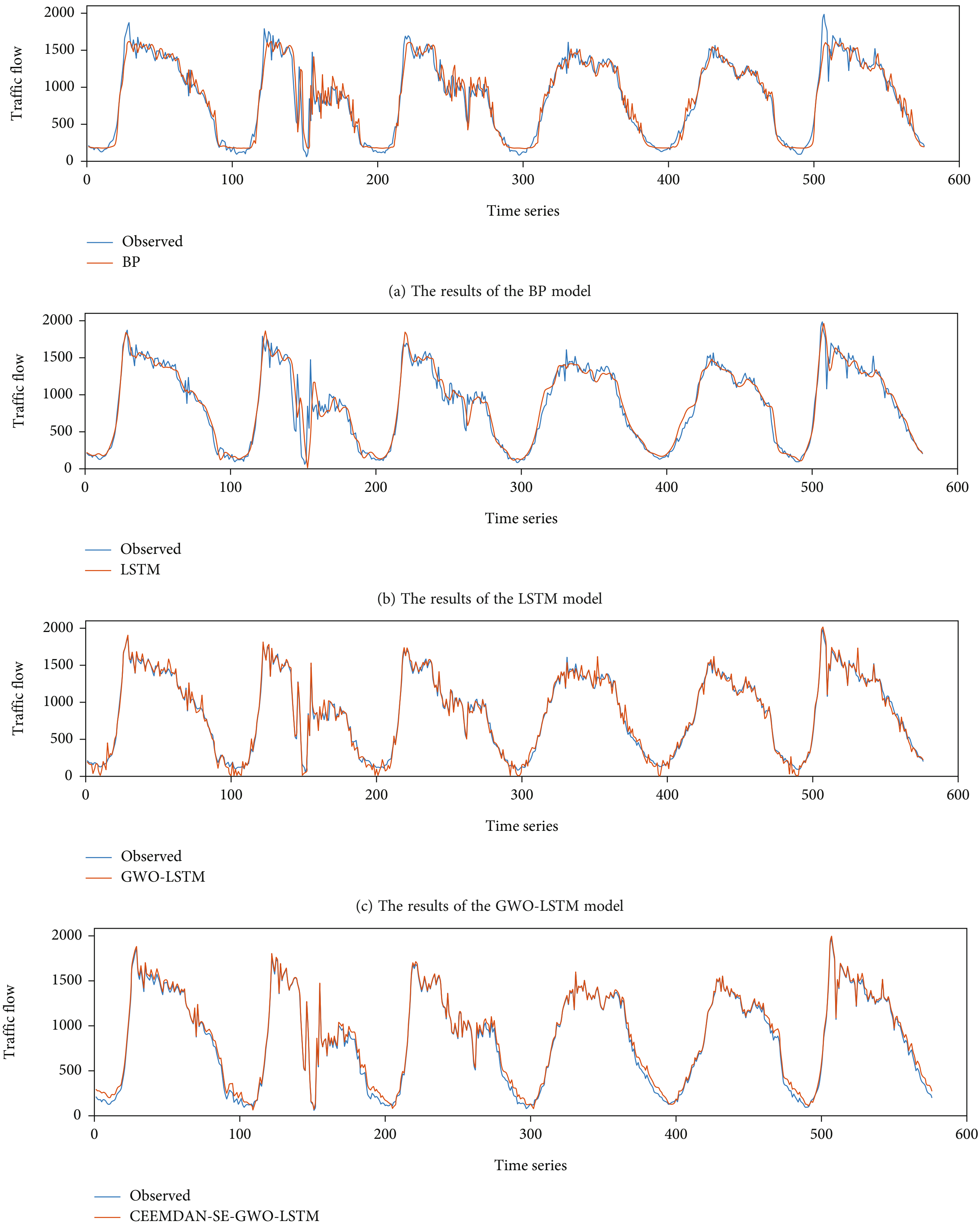


Figure 9: The prediction results of the subsequences.

It can make better use of the information of each subsequence to obtain an integrated model with a stable trend response, adapting to extreme values, overcoming delay effects, and practical significance. Finally, we integrate the prediction results of different models into Figure 10(e).

3.5. Analysis of Interval Prediction Results. First, we use KDE to estimate the distribution functions of the prediction errors of different models in the previous section and then calculate the error intervals under different confidence levels through ICDF. Table 5 is the error interval of different



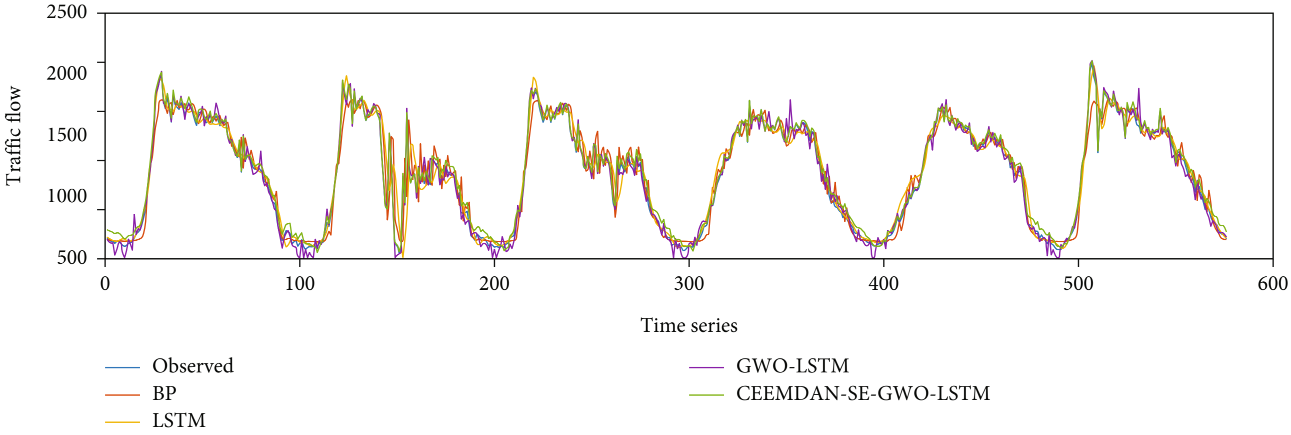
(a) The results of the BP model

(b) The results of the LSTM model

(c) The results of the GWO-LSTM model

(d) The results of the CEEMDAN-SE-GWPO-LSTM model

Figure 10: Continued.



(e) Comparison of different models

Figure 10: The prediction results of traffic flow prediction.

Table 5: Error fluctuation intervals for different models.

Confidence	BP	LSTM	GWO-LSTM	CEEDMAN-SE-GWO-LSTM
95%	(-275.664, 288.577)	(-263.724, 224.41)	(-111.734, 113.799)	(-96.7679, 15.1717)
90%	(-212.354, 200.558)	(-189.884, 154.638)	(-79.4818, 93.3297)	(-89.2135, 6.25768)
80%	(-156.438, 130.56)	(-140.351, 98.9233)	(-57.4794, 68.4463)	(-79.7756, -0.818)

models under different confidence levels. From a longitudinal perspective, the error interval of any model increases with the increase of confidence. As the model continues to be optimized, horizontally, our proposed model performs the best with any error interval at any confidence level.

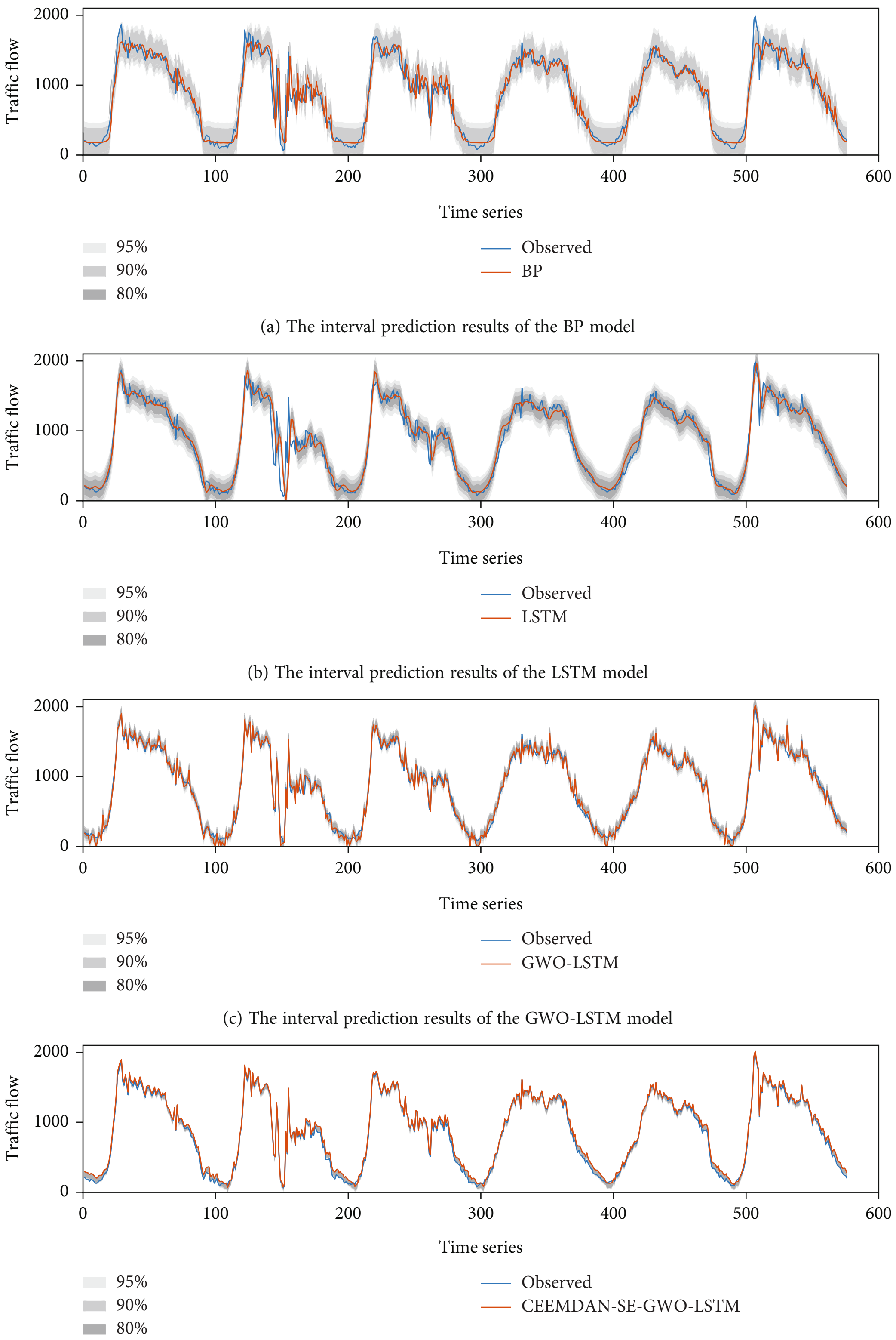
Figure 11 shows that in the BP and LSTM models, the prediction error is large, and the prediction is hysteretic, which leads to a traffic flow prediction error of 200-300 when the confidence level is 80%. This is very disadvantageous for traffic planning dynamic planning. At the high confidence level of 80% in our proposed model, the error every fifteen minutes can be controlled within 100, and it can be seen from the figure that most of the true observations are within the prediction interval. Even some extreme fluctuations are included, which shows that the interval prediction proposed in this paper can improve the accuracy of STTF.

4. Conclusion and Discussion

In this paper, we proposed a method to solve accurate short-term predictions for signals, using real datasets for model validation. The method is generally integrated into two parts: CEEMDAN-SE and GWO-LSTM. First, we performed data cleaning on the traffic flow dataset by removing outliers and invalid values and then performing interpolation to ensure data consistency. Secondly, the CEEMDAN method is used to deconstruct the time series data of traffic flow. In order to prevent excessive computational scale, we introduced SE to measure the nonlinear complexity of the IMF subsequence processed by CEEMDAN to reduce the dimension of the IMF subsequence. The data is then fed into the optimised GWO-LSTM model. GWO minimises its loss function by continuously and dynamically optimising the hyperparameters during training with different sequences to achieve the desired accuracy. Finally, the component prediction results are integrated to obtain the predicted value, and the quantitative calculation of the evaluation index is carried out. The results evidence how our proposed ensemble model performs the best compared to other commonly used models. In order to improve the generalization of our model, we propose an interval prediction model based on KDE theory, which can increase the accuracy and be more practical. Our method can accurately predict extreme values with a rapid response to sudden changes in short-term trends. After the modal decomposition, the component features can be better utilised so that the final predicted value and its derivative will be smoother and have more practical application significance. The research shows that the ensemble model proposed by us can improve STFF prediction accuracy to a certain extent, which can provide a reference for related research.

Our work also presents certain limitations. More methods to evaluate the importance of the variables for feature selection might be required. Some scholars use a random forest algorithm (RF) to evaluate it, which more rigorously demonstrates the excellence of their model. Furthermore, both CEEMDAN-SE and GWO-LSTM could be improved if the two parts are combined well. We initial GWO model by random values in different values. It might be better to fix the values adapted to CEEMDAN-SE.

Thus, our future work is to solve the spiking phenomenon after CEEMDAN decomposition; the reason is that the decomposition is not thorough enough and has



(a) The interval prediction results of the BP model

(b) The interval prediction results of the LSTM model

(c) The interval prediction results of the GWO-LSTM model

(d) The interval prediction results of the CEEMDAN-SE-GWO-LSTM model

FIGURE 11: Interval prediction results based on different models.

pseudomodalities. Using the IMF component of the original noise signal as the noise for each calculation of the IMF may better avoid pseudomodal phenomena and reduce unnecessary components to improve the input data quality of subsequent prediction models. For better prediction, other better time-series models can be applied to form a combined model. For instance, when given better time-series model X, scholars could consider using the GWO model to optimize the time-series model, and we hope to be able to consider other dimension features to get better prediction performance, find the most suitable combination, and use a machine learning model like random forest to evaluate.

Data Availability

The data used to support the findings of this study have been deposited in https://github.com/Bobbed1999/Short-Term-Traffic-Flow-Prediction.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Wenzheng Zhao designed the study, performed data analysis and experimental research, conducted experiments on the proposed model, data curation and visualization of experimental results, and first draft review and editing. Yuer Yang conducted experiments with the part of proposed model and collated the experimental results and wrote part of the first draft. Zihao Lu collected related work and wrote the introduction.

Acknowledgments

Thanks to all authors for their contributions in this work and the related work [33] of the lead author contributed to this paper. Thanks to the anonymous reviewers for their insightful comments.

References

[1] J. Zheng and M. Huang, "Traffic flow forecast through time series analysis based on deep learning," *IEEE Access*, vol. 8, pp. 82562–82570, 2020.

[2] C. Ravi, A. Tigga, G. T. Reddy, S. Hakak, and M. Alazab, "Driver identification using optimized deep learning model in smart transportation," *ACM Transactions on Internet Technology*, 2022.

[3] X. Pan, X. Cai, K. Song, T. Baker, T. R. Gadekallu, and X. Yuan, "Location recommendation based on mobility graph with individual and group influences," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2022.

[4] K. Yu, L. Lin, M. Alazab, L. Tan, and B. Gu, "Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4337–4347, 2021.

[5] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2021.

[6] N. Kumar, S. S. Rahman, and N. Dhakad, "Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4919–4928, 2021.

[7] D. Zhang, J. Wang, H. Fan, T. Zhang, J. Gao, and P. Yang, "New method of traffic flow forecasting based on quantum particle swarm optimization strategy for intelligent transportation system," *International Journal of Communication Systems*, vol. 34, no. 1, p. e4647, 2021.

[8] S.-Y. Chen and W.-Y. Chou, "Short-term traffic flow prediction using emd-based recurrent hermite neural network approach," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 1821–1826, Anchorage, AK, USA, 2012.

[9] J. Liu, N. Wu, Y. Qiao, and Z. Li, "Short-term traffic flow forecasting using ensemble approach based on deep belief networks," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[10] L. Wenqi, Y. Rui, Z. Yi, B. Ran, and G. Yuanli, "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.

[11] T. Zhou, G. Han, X. Xuemiao et al., "_δ_ -agree AdaBoost stacked autoencoder for short-term traffic flow forecasting," *Neurocomputing*, vol. 247, pp. 31–38, 2017.

[12] G. Yu and C. Zhang, "Switching ARIMA model based forecasting for traffic flow," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. ii–429, Montreal, QC, Canada, 2004.

[13] S. Vasantha Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *European Transport Research Review*, vol. 7, no. 3, pp. 1–9, 2015.

[14] C. Chen, H. Jianming, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with ARIMA-GARCH model," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 607–612, Baden-Baden, Germany, 2011.

[15] S. V. Kumar, "Traffic flow prediction using Kalman filtering technique," *Procedia Engineering*, vol. 187, pp. 582–587, 2017.

[16] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

[17] Y. N. Yang and H. P. Lu, "Short-term traffic flow combined forecasting model based on SVM," in *2010 International Conference on Computational and Information Sciences*, pp. 262–265, Chengdu, China, 2010.

[18] M. Duan, "Short-time prediction of traffic flow based on PSO optimized SVM," in *2018 international conference on intelligent transportation, big data & smart city (ICITBS)*, pp. 41–45, Xiamen, China, 2018.

[19] I. Alam, D. M. Farid, and R. J. Rossetti, "The prediction of traffic flow with regression analysis," in *Emerging Technologies in Data Mining and Information Security*, pp. 661–671, Springer, 2019.

[20] W. Zhang, Y. Yinghao, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning," *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1688–1711, 2019.

[21] Q. Zhaowei, L. Haitao, L. Zhihui, and Z. Tao, "Short-term traffic flow forecasting method with MB-LSTM hybrid network," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[22] D. Ma, X. Song, and P. Li, "Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter- and intra-day traffic patterns," *IEEE*

Transactions on Intelligent Transportation Systems, vol. 22, no. 5, pp. 2627–2636, 2021.

[23] 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, pp. 114496–114507, 2019.

[24] 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 of London. Series A: mathematical, physical and engineering sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[25] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4144–4147, Prague, Czech Republic, 2011.

[26] S. Pincus, "Approximate entropy (apen) as a complexity measure," *Journal of Nonlinear Science*, vol. 5, no. 1, pp. 110–117, 1995.

[27] K. Wang, D. Niu, L. Sun et al., "Wind power short-term forecasting hybrid model based on CEEMD-SE method," *Processes*, vol. 7, no. 11, p. 843, 2019.

[28] Z. Tian, "Approach for short-term traffic flow prediction based on empirical mode decomposition and combination model fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5566–5576, 2021.

[29] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 278, no. 6, pp. H2039–H2049, 2000.

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

[31] S. Prasanth, U. Singh, A. Kumar, V. A. Tikkiwal, and P. H. Chong, "Forecasting spread of covid-19 using google trends: a hybrid GWO-deep learning approach," *Chaos, Solitons & Fractals*, vol. 142, article 110336, 2021.

[32] W. H. K. Lam and M. L. Tam, "Risk analysis of traffic and revenue forecasts for road investment projects," *Journal of Infrastructure Systems*, vol. 4, no. 1, pp. 19–27, 1998.

[33] W. Zhao, "Accurate non-stationary short-term traffic flow prediction method," 2022, https://arxiv.org/abs/2205.00517.