FINAL REPORT

PROJECT K3

AUGUST 2022

Traffic Congestion Identification and Prediction Based on Image Processing and Deep Learning Methods

Robert Whalin, Ph.D., Jackson State University
Guojing Hu, Ph.D., Jackson State University

STRIDE | Southeastern Transportation Research, Innovation, Development and Education Center
UF | Transportation Institute UNIVERSITY of FLORIDA

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. Project K3	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Traffic Congestion Identification and Prediction Based on Image Processing and Deep Learning Methods		**5. Report Date** 8/2/2022 **6. Performing Organization Code**
7. Author(s) Robert Whalin, Ph.D., Jackson State University Guojing Hu, Ph.D., Jackson State University		**8. Performing Organization Report No.** STRIDE Project K3
9. Performing Organization Name and Address *Jackson State University, 1400 J.R. Lynch Street, P.O. Box 17068, Jackson, MS* 39217-0168		**10. Work Unit No.** **11. Contract or Grant No.** Funding Agreement Number 69A3551747104
12. Sponsoring Agency Name and Address *University of Florida Transportation Institute/* Southeastern Transportation Research, Innovation, Development and Education Center (STRIDE) 365 Weil Hall, P.O. Box 116580 Gainesville, FL 32611 *U.S Department of Transportation/Office of Research, Development & Tech* 1200 New Jersey Avenue, SE Washington, DC 20590		**13. Type of Report and Period Covered** 1/1/2020 to 8/2/2022 **14. Sponsoring Agency Code**

15. Supplementary Notes

16. Abstract - Short-term traffic flow/speed prediction is a significant and challenging research topic as it is closely related to the application of intelligent transportation systems. Due to the variable and random characteristics of the transportation system, raw traffic flow/speed data often contains noise, and predicting it directly may reduce the accuracy and effectiveness of the prediction models. Therefore, a novel hybrid method is established in this project which combines denoising schemes and deep learning model to improve prediction accuracy. The time series denoising schemes include two parts: the complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN), and wavelet packet decomposition (WPD). Firstly, the raw traffic flow data are decomposed by CEEMDAN to gain intrinsic mode functions (IMFs) and a residual. Then the WPD algorithm conducts a secondary decomposition to part of the IMFs that still contains complex and high-frequency component signals of the volume series. Traditional deep learning models for time series prediction are usually based on recurrent neural networks, which have the disadvantages of being time-consuming and prone to problems such as gradient explosions or gradient vanishing. This project uses a relatively new approach, called deep echo state network (DeepESN), which can obtain good prediction results while circumventing the above problems. In the experiment, to investigate the prediction performance of the proposed CEEMDAN-WPD-DeepESN model, the LSTM, CEEMDAN-LSTM, CEEMDAN-WPD-LSTM, DeepESN, CEEMDAN-DeepESN, CEEMDAN-WPD-DeepESN and CEEMDAN-WPD6-DeepESN models are considered to be comparison models. In addition, network-wide traffic state forecasting is a more challenging task, due to the complex spatial dependencies and the time-varying traffic patterns on road networks. To address this challenge, we treat the traffic network as a graph. Combining graph convolution techniques with deep learning methods to capture the spatial and temporal dependencies of road networks, and eventually improve the prediction performance of network-wide traffic states.

17. Key Words	18. Distribution Statement
secondary decomposition, DeepESN, traffic state prediction, graph convolution	No restrictions

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
N/A	N/A	43 pages	N/A

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

ACKNOWLEDGEMENT OF SPONSORSHIP AND STAKEHOLDERS

This work was sponsored by a contract from the Southeastern Transportation Research, Innovation, Development and Education Center (STRIDE), a Regional University Transportation Center sponsored by a grant from the U.S. Department of Transportation's University Transportation Centers Program.

Funding Agreement Number - 69A3551747104

LIST OF AUTHORS

Lead PI:

Robert W. Whalin, Ph.D., P.E.
Jackson State University
robert.w.whalin@jsums.edu
https://orcid.org/0000-0002-8712-9434

Additional Researchers:

Guojing Hu, Ph.D.
Jackson State University
huguojinghu@foxmail.com
https://orcid.org/0000-0001-6246-1513

TABLE OF CONTENTS

DISCLAIMER ii
ACKNOWLEDGEMENT OF SPONSORSHIP AND STAKEHOLDERS ii
LIST OF AUTHORS iii
LIST OF FIGURES vi
LIST OF TABLES vii
ABSTRACT viii
EXECUTIVE SUMMARY ix
1.0 TASK 1: SHORT-TERM TRAFFIC FLOW PREDICTION BASED ON SECONDARY HYBRID DECOMPOSITION AND DEEP ECHO STATE NETWORKS 10
1.1 INTRODUCTION 10
1.1.1 Objective 10
1.1.2 Scope 10
1.2 LITERATURE REVIEW 10
1.2.1 Statistical Methods 10
1.2.2 Data-driven Machine Learning Methods 10
1.2.3 Hybrid Methods 11
1.3 METHODOLOGY 12
1.3.1 Framework of Modelling 12
1.3.2 Empirical Mode Decomposition Family 13
1.3.3 Wavelet Packet Decomposition (WPD) 15
1.3.4 Deep Echo State Network (DeepESN) 16
1.3.5 Evaluation Metrics 19
1.4 RESULT 20
1.4.1 Data Set 20
1.4.2 Forecasting Results 21
1.5 CONCLUSION 31
2.0 TASK 2: SHORT-TERM TRAFFIC SPEED PREDICTION ON A ROAD NETWORK 32
2.1 INTRODUCTION 32
2.1.1 Objective 32
2.1.2 Scope 32

2.2 LITERATURE REVIEW 32
2.3 METHODOLOGY 33
2.3.1 Graph Convolutional Long Short Term Memory (GC-LSTM) 33
2.3.2 Graph Convolutional Neural Network with Gated Recurrent Unit (GCN-GRU) 34
2.3.3 Loss Function 35
2.4 RESULT 36
2.4.1 Data Set 36
2.4.2 Simulation Environment 37
2.4.3 Model Parameter Selection 37
2.4.4 Forecasting Results 37
2.5 CONCLUSION 38
3.0 RECOMMENDATIONS 38
4.0 REFERENCE LIST 40

LIST OF FIGURES

Figure 1 Framework of the hybrid CEEMDAN-WPD-DeepESN model. 13

Figure 2 The three-layer decomposition trees of WD and WPD. 16

Figure 3 A generic ESN model. 17

Figure 4 Illustration of a deep echo state network. 18

Figure 5 The selected sensor A in the city of Atlanta. 20

Figure 6 Original traffic flow time series of sensor A. 21

Figure 7 The original traffic flow series decomposed to IMF1-IMF11 and residual by the CEEMDAN method. 22

Figure 8 IMF1 decomposed to IMF1_WPD1-IMF1_WPD8 components by the WPD method. 24

Figure 9 The prediction result of the test data. 25

Figure 10 Prediction results of different models. 30

Figure 11 The architecture of the GC-LSTM model. 34

Figure 12 The architecture of the GCN-GRU model. 35

Figure 13 The selected 501 sensors in the city of Atlanta. 36

LIST OF TABLES

Table 1 Hurst Exponent of each component decomposed by CEEMDAN. 23

Table 2 Selection of hypermeters used in DeepESN model. 25

Table 3 Performance evaluation results of different models. 30

Table 4 Comparison results of different models. 37

ABSTRACT

Short-term traffic flow/speed prediction is a significant and challenging research topic as it is closely related to the application of intelligent transportation systems. Due to the variable and random characteristics of the transportation system, raw traffic flow/speed data often contains noise, and predicting it directly may reduce the accuracy and effectiveness of the prediction models. Therefore, a novel hybrid method is established in this project which combines denoising schemes and deep learning model to improve prediction accuracy. The time series denoising schemes include two parts: the complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN), and wavelet packet decomposition (WPD). Firstly, the raw traffic flow data are decomposed by CEEMDAN to gain intrinsic mode functions (IMFs) and a residual. Then the WPD algorithm conducts a secondary decomposition to part of the IMFs that still contains complex and high-frequency component signals of the volume series. Traditional deep learning models for time series prediction are usually based on recurrent neural networks, which have the disadvantages of being time-consuming and prone to problems such as gradient explosions or gradient vanishing. This project uses a relatively new approach, called deep echo state network (DeepESN), which can obtain good prediction results while circumventing the above problems. In the experiment, to investigate the prediction performance of the proposed CEEMDAN-WPD-DeepESN model, the LSTM, CEEMDAN-LSTM, CEEMDAN-WPD-LSTM, DeepESN, CEEMDAN-DeepESN, CEEMDAN-WPD-DeepESN and CEEMDAN-WPD6-DeepESN models are considered to be comparison models.

In addition, network-wide traffic state forecasting is a more challenging task, due to the complex spatial dependencies and the time-varying traffic patterns on road networks. To address this challenge, we treat the traffic network as a graph. Combining graph convolution techniques with deep learning methods to capture the spatial and temporal dependencies of road networks, and eventually improve the prediction performance of network-wide traffic states.

Keywords (up to 5):
Secondary decomposition, DeepESN, traffic state prediction, graph convolution

EXECUTIVE SUMMARY

Due to the wide application in public transportation scheduling, route planning and signal optimization to alleviate traffic congestion, short-term traffic flow/speed prediction has been a significant consideration in the intelligent transportation system. If there is an accurate short-term traffic state prediction method, we can accurately describe and predict the evolution of traffic state over time through historical data, so that it would be possible to carry out effective traffic management and control methods, evacuate congestion, optimize routes, etc. Therefore, this project aims to propose effective models for short-term traffic state prediction.

Task 1 focuses on traffic forecasting on key roads. Thus, a sensor on I-285, considered one of the busiest and most congested roads in Atlanta, has been selected for this study. We propose a novel hybrid model (CEEMDAN-WPD-DeepESN) that first decomposes the original traffic state series with noise, then processes the random component, and finally predicts each sub-series separately and aggregates to get the final results. The experimental results demonstrate that the proposed CEEMDAN-WPD-DeepESN model have much superior performance on both efficiency and accuracy.

Task 2 focuses on traffic forecasting on the road network. Thus, 501 sensors among the city of Atlanta have been selected for this study. As we know, the traffic states on these 501 sensors are not isolated, they influence each other. But which sensors have influence, how and to what extent is a complicated issue. Therefore, when predicting the network-wide traffic state, it is also questionable whether to and how to consider the influence between the sensors. In this study, we apply the graph convolutional technique to mine the spatial relationships of traffic speed over multiple sensors, and then feed the output the deep learning models (e.g., LSTM, GRU, DeepESN) to extract temporal features embedded in traffic speed. The experiments produce two interesting findings: (1) the GC-LSTM model achieves better prediction performance than the individual LSTM model, but the GC-DeepESN model yields worse prediction performance than the individual DeepESN model. Therefore, when modeling the road network, its spatial structure needs to be treated with caution. (2) The DeepESN model produces the smallest MSE, even smaller than that of the GC-LSTM and GCN-GRU models. This phenomenon proves the superiority of the DeepESN model in predicting traffic network states.

The findings derived from Task 1 and Task 2 can help traffic managers to quickly and accurately predict the future traffic status of congested roads so that controlling measures can be taken in advance to prevent the spread of congestion.

1.0 TASK 1: SHORT-TERM TRAFFIC FLOW PREDICTION BASED ON SECONDARY HYBRID DECOMPOSITION AND DEEP ECHO STATE NETWORKS

1.1 INTRODUCTION

Short-term traffic volume data are often corrupted by local noises, which may influence the prediction accuracy of short-term traffic volumes. Complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) and wavelet packet decomposition (WPD) analysis are used in this study to decompose the original data into multiple components such that the deep echo state network (DeepESN) model can then be applied to the denoised data, and the forecasting accuracy can be improved.

1.1.1 Objective

This project proposes a hybrid model to forecast the short-term traffic flow. The objectives are to: (1) Extract and handle the nonstationary part of the traffic flow series. (2) Improve the accuracy and efficiency of short-term traffic flow forecasting.

1.1.2 Scope

This task focuses on sensor data in the city of Atlanta. The data attributes include traffic volume, speed, and occupancy. This project conducts traffic volume predictions and traffic speed predictions.

1.2 LITERATURE REVIEW

Significant efforts have been devoted to developing accurate and efficient methodologies for short-term traffic flow prediction. In general, there are three categories of traffic flow forecasting methods: (1) traditional statistical methods, (2) computational intelligence-based methods, and (3) hybrid methods.

1.2.1 Statistical Methods

The main idea of statistical methods is to regard traffic flow as time series data and employ statistical analysis tools as the basic model to investigate the intrinsic relationship between past and future time series values. The statistical methods include Kalman filter (Okutani and Stephanedes 1984, Whittaker et al. 1997, Stathopoulos and Karlaftis 2003, Ma et al. 2017), autoregressive integrated moving average (ARIMA) and its variants (Ahmed and Cook 1979, Levin et al. 1980, Hamed et al. 1995, Voort et al. 1996, Lee et al. 1999, Williams et al. 1998, Williams 2001, Williams and Hoel 2003, Chandra and Deek 2009), and spectral analysis (Nicholson and Swann 1974, Tchrakian 2012). Among these methods, ARIMA family has been the most popular and widely applied in traffic flow prediction due to its powerful linear fitting ability (Yao et al. 2020). However, due to the large amount of noise in traffic data (Yu et al. 2021), transportation system is considered as a nonlinear and nonstationary system, which poses substantial challenges to ARIMA methods based on the linear assumption.

1.2.2 Data-driven Machine Learning Methods

To address the nonlinear traffic flow forecasting problem, many data-driven machine learning methods have been developed, such as support vector regression (SVR) (Wu et al. 2004,

Manoel et al. 2009, Zhang and Liu 2009, Zhang and Xie 2007, Asif et al. 2013), artificial neural network (ANN) (Affiliation et al. 2015, Rilett and Park 2001, Jiang and Adeli 2005) and its variant - recurrent neural network (RNN) (Vlahogianni et al. 2005). In recent years, RNN has made great strides in traffic prediction for its capability of capturing temporal dependence of time series data. The two most popular RNN architectures for traffic state prediction are long short-term memory (LSTM) neural network (Ma et al. 2015, Tian et al. 2018) utilizing three gates: input, output, and forget gates, and gated recurrent unit (GRU) neural network (Jeong et al. 2021, Fu et al. 2016) with two gates: update and reset gates. However, due to the training process of backpropagation, the problem of vanishing gradient exists in LSTM and GRU. In addition, LSTM and GRU architectures adopt a gradient descent method to adapt input, recurrent, output connections, which renders low computational efficiency. To overcome the vanishing gradient and low efficiency problems, echo state network (ESN) has been noticed as a novel approach to model RNNs in recent years (Han and Xu 2017, Lin et al. 2009, Shi and Han 2007, Wang et al. 2019). The advantage of ESN structure lies in that only the output weights need to be trained, and the training method is linear regression rather than backpropagation. These characteristics make ESN a highly efficient neural network for traffic flow prediction. A first attempt to apply ESN to traffic flow prediction was made by An, Song and Zhao (2011), they found that the ESN-based short-term traffic flow prediction algorithm can achieve higher accuracy than the feed forward neural network. Then, a more complete study was conducted by Ser et al. (2019), they verified the advantages of the stacking ensembles of ESNs in improving traffic prediction performance and reducing training time. In 2018, deep echo state network (DeepESN) was designed by applying the deep learning framework to ESN (Gallicchio et al. 2018). Ser et al. (2020) continued to explore the suitability of DeepESN for short-term traffic forecasting and concluded that DeepESN achieved more accurate traffic forecasts than other considered counterparts.

1.2.3 Hybrid Methods

Recently, to further enhance the accuracy of traffic prediction, some researchers have applied a data denoising process before the application of machine learning/deep learning networks. The commonly used denoising techniques include empirical mode decomposition (EMD) family (Pholsena et al. 2019, Tang et al. 2019, Tian et al. 2019, Kim et al. 2019, Wang et al. 2014, Jiang et al. 2014) and wavelet-based decomposition method (Zhang et al. 2019, Liu et al. 2019, Zhang et al. 2019, Sun et al. 2019, Sun et al. 2015). The hybrid prediction methods that combine data denoising and machine learning/deep learning models have been proven to have better performance than single models (Tang et al. 2019, Kashi and Akbarzadeh 2019, Wei and Chen 2012, Li et al. 2018, Diao et al. 2019). However, most of the existing hybrid methods adopt the traditional backpropagation-based deep learning models to forecast the denoised traffic data. As we mentioned before, such backpropagation-based methods may suffer from vanishing gradient and high computational cost, and DeepESN is an effective alternative. To the best of our knowledge, Kim and King are the only researchers to have developed a hybrid approach that uses DeepESN for time series prediction, the hybrid approach was verified to be efficient

and accurate in forecasting S&P 500 data, parking data, chaos data, temperature data and bike sharing data (Kim and King 2020). In this study, we will continue along this research path by proposing a novel secondary hybrid model for short-term traffic flow prediction. The major contributions of this work are summarized as follows:

(1) To deal with the non-stationary and uncertainty of the original traffic flow data, two different decomposition techniques, namely complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) and wavelet packet decomposition (WPD), are used to decompose the traffic flow series into multiple predictable sub-series.

(2) This is the first attempt in the field of traffic prediction to combine two decomposition techniques with DeepESN architecture (CEEMDAN-WPD-DeepESN) for fast and accurate prediction.

(3) The proposed model is comprehensively compared, not only with the LSTM neural network, but also with LSTM-based ensemble models.

1.3 METHODOLOGY

1.3.1 Framework of Modelling

The proposed hybrid approach for short-term traffic flow prediction has two main stages. Firstly, a secondary decomposition method (CEEMDAN-WPD) is applied to decompose the traffic flow series. The DeepESN model is then used to forecast all the sub-series generated by CEEMDAN-WPD. Finally, the prediction results of each sub-series are aggregated to reconstruct the traffic flow series. Figure 1 illustrates the proposed hybrid approach. The modelling steps are as follows.

(1) Use CEEMDAN to decompose the original traffic flow series into a set of intrinsic mode functions (IMF1, IMF2…) and a residual. The modeling details of the CEEMDAN algorithm are illustrated in Section 1.3.2.

(2) Apply a three-layer WPD to further decompose IMF1 into eight sub-series (IMF1-WPD1, IMF1-WPD2, …, IMF1-WPD8). The modeling details of the WPD algorithm are illustrated in Section 1.3.3.

(3) Utilize DeepESN to predict each sub-series, and sum the predicted results of all sub-series to obtain the final forecasting results. The modeling details of the DeepESN algorithm are illustrated in Section 1.3.4.

(4) To investigate the prediction performance of the proposed CEEMDAN-WPD-DeepESN framework, the same traffic flow series are also forecasted by other models, including DeepESN, CEEMDAN-DeepESN, LSTM, CEEMDAN-LSTM, CEEMDAN-WPD-LSTM.

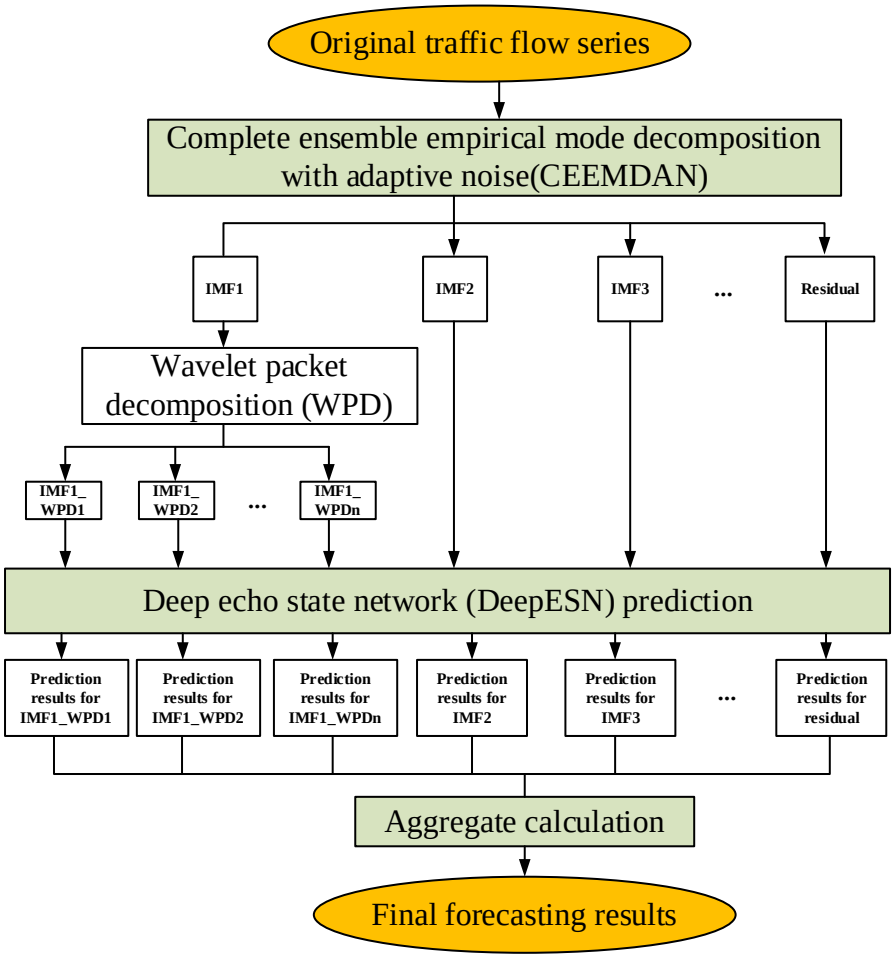


FIGURE 1 FRAMEWORK OF THE HYBRID CEEMDAN-WPD-DEEPESN MODEL.

1.3.2 Empirical Mode Decomposition Family

(1) Empirical mode decomposition (EMD)

Empirical mode decomposition (EMD) is an adaptive time series decomposition technique, which utilizes the Hilbert-Huang transform (HHT) to sift the nonlinear signal until it becomes stationary (Huang et al. 1998). Due to the ability of EMD in analyzing nonlinear and non-stationary signals (Liu et al. 2012, Wei and Chen 2012) and the nonlinear and non-stationary nature of traffic data, the EMD has been widely used in traffic flow/speed prediction.

Given an original traffic flow/speed series $x(t)$, EMD decomposes the original data into a collection of IMFs and a residual according to the local features of the data itself, such as local maxima, local minima, and zero-crossing. The decomposition procedure of EMD is as follows:

Step 1: Identify the local maxima and minima of the original data $x(t)$, and fit the local extreme with cubic spline function to generate the upper envelope $u(t)$ and lower envelope $l(t)$.

Step 2: Calculate the mean of the upper and lower envelopes $m(t) = \frac{u(t)+ l(t)}{2}$.

Step 3: Subtract the mean from the original data to get a new sequence $d(t) = x(t) - m(t)$.

Step 4: If $d(t)$ and $m(t)$ satisfy one of the stopping criteria, then the first IMF $c_1(t) = d(t)$, the first residual $r_1(t) = x(t) - d(t)$. The stopping criteria are: i) $d(t)$ approaches zero; ii) the number of extreme and zero-crossing value of $d(t)$ either equal or differ at most by one.

Step 5: Else, replace $x(t)$ with $d(t)$, and repeat Steps 1-4 to obtain the $c_1(t)$ and $r_1(t)$.

Step 6: Regard $r_1(t)$ as the new time series, repeat Steps 1-5 n times until the residual $r_n(t)$ becomes a constant or a monotonic function that cannot be further decomposed into IMFs.

Finally, the original data can be expressed as the sum of IMFs $c_i(t), i = 1,2, \dots, n$ and a residual:

$$x(t) = \sum_{i=1}^{n} c_i(t) + r_n(t)$$

Where n represents the number of IMFs.

(2) Ensemble empirical mode decomposition (EEMD)

EMD may fail in the denoising procedure when the to-be denoised data cannot meet certain prerequisites (Tong et al. 2012). The most serious problem in EMD is the mode mixing phenomenon. Mode mixing refers to the situation where the IMF produced by EMD decomposition has components with dramatically different frequencies. In order to alleviate the mode mixing problem, Wu and Huang (2009) proposed a noise-assisted data analysis method called ensemble empirical mode decomposition (EEMD). The white noise can provide uniformly distributed scales in the time-frequency space. The intrinsic oscillations in the signal with different scales will automatically correlate with a reference grid of similar scales provided by white noise. Therefore, the intrinsic local oscillations can be adaptively filtered to the appropriate scales by the natural filter bank of the EMD in relation to the added uniformly distributed white noise. The mode mixing problem is solved by this elegant use of noise. The EEMD algorithm can be described as follows:

Step 1: Add a collection of white noise to the original data: $x^i(t) = x(t) + \beta\varepsilon^i(t)$, where $\varepsilon^i(t)$ $(i = 1,2, \dots, I)$ is a zero mean unit variance white noise series, I is the total number of trials, and $\beta > 0$.

Step 2: Decompose the noise-aided data $x^i(t)$ by EMD, obtaining the IMF $c_k^i(t)$ $(k = 1,2, \dots, K \text{ is the mode})$, where K is the total number of modes, and the residual can be expressed as $r^i(t) = x^i(t) - \sum_{k=1}^{K} c_k^i(t)$

Step 3: Obtain the kth $\widetilde{IMF}$ of EEMD as $\widetilde{c_k}(t) = \frac{1}{I}\sum_{i=1}^{I} c_k^i(t)$, and the residual is $\widetilde{r_K}(t) = \frac{1}{I}\sum_{i=1}^{I} r^i(t)$.

(3) Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN)

It can be noticed that in EEMD, every $x^i(t)$ is decomposed independently and a residual is obtained at each stage, with no connections between different trials. This situation can result in some EEMD shortcomings: i) decomposition is not complete, ii) different realization of signal plus noise might yield different number of modes. In order to further alleviate the mode mixing and reconstruction error problems, Torres et al. (2011) proposed the complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) by adding the adaptive white Gaussian noise at each decomposition. The computation of the CEEMDAN can be described as follows:

Step 1: Add a collection of white Gaussian noise series to the original data: $x^i(t) = x(t) + w_0\varepsilon^i(t)$, where $\varepsilon^i(t)$ $(i = 1,2, \dots, I)$ is the white Gaussian noise series, I is the total number of trials, and w_0 is a noise coefficient.

Step 2: Decompose the noise-aided data $x^i(t)$ by EMD, obtaining the first IMF $c_1^i(t)$.

Step 3: Obtain the first $\overline{IMF}$ of CEEMDAN as $\overline{c_1}(t) = \frac{1}{I}\sum_{i=1}^{I} c_1^i(t)$.

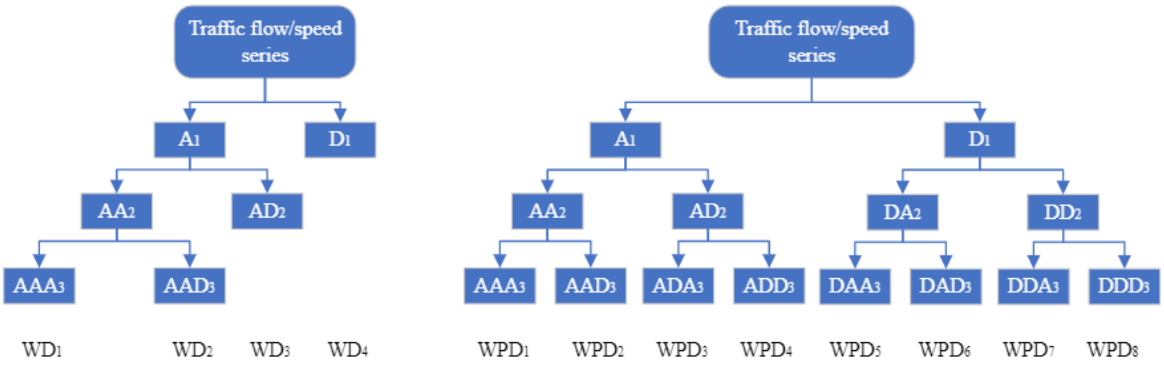
Step 4: Calculate the kth residual as $\overline{r_k}(t) = x(t) - \overline{c_k}(t)$.

Step 5: Decompose the noise-added residual $\overline{r_k}(t) + w_k E_k(\varepsilon^i(t))$ to obtain the $(k+1)$th $\overline{IMF}$ as $\overline{c_{k+1}}(t) = \frac{1}{I}\sum_{i=1}^{I} E_k(\overline{r_k}(t) + w_k E_k(\varepsilon^i(t)))$, where $E_k(\cdot)$ is the operator that produces the kth mode obtained by EMD.

Step 6: Repeat step 4-step 5 for the remaining $\overline{IMF}$s until the obtained residual cannot be further decomposed by EMD, and the final residual can be expressed as $\overline{r_K}(t) = x(t) - \sum_{k=1}^{K}\overline{c_k}(t)$ $(k = 1,2, \dots, K \text{ is the mode})$, where K is the total number of $\overline{IMF}$s, and the original data also can be expressed as the sum of $\overline{IMF_k}$ plus a residual: $x(t) = \sum_{k=1}^{K}\overline{c_k}(t) + \overline{r_K}(t)$.

1.3.3 Wavelet Packet Decomposition (WPD)

Wavelet decomposition (WD) (Aghajani et al. 2016) is widely used as an effective method to decompose the non-stationary signals into multiple frequencies. While WD can only decompose appropriate components, wavelet packet decomposition (WPD), as a special WD method, can decompose both the appropriate components and the detailed components (Zhao et al. 2010). For example, the three-layer binary tree of WD obtains four components (WD1-WD4), and the three-layer binary tree of WPD obtains eight components (WPD1-WPD8), as seen in Figure 2.



(a) WD (b) WPD

FIGURE 2 THE THREE-LAYER DECOMPOSITION TREES OF WD AND WPD.

The WPD includes two types: continuous wavelet transform (CWT), and discrete wavelet transform (DWT). The CWT can be defined as:

$$CWT_f(a,b) = \langle f(t), \Psi_{a.b}(t) \rangle = \int_{-\infty}^{+\infty} f(t)\Psi^*((t-b)/a)/\sqrt{a}\, dt$$

Where $f(t)$ is the signal, $\Psi(t)$ is the mother wavelet function, $*$ denotes the complex conjugate, a and b represent the scale coefficient and translation coefficient, respectively. In CWT, the scale and translation coefficients are continuous, which makes CWT quite slow. Therefore, CWT is seldom employed for prediction. Instead, the wavelet of DWT is scaled and translated using certain scales and positions that can provide the necessary information while reducing the calculation cost. In detail, DWT uses scale and position values based on powers of two, and the scale coefficient a is discretely expressed as $a = 2^j$ and translation coefficient b is discretely described as $b = k \cdot 2^j$, where j and k are integers that control the scale and translation, respectively.

1.3.4 Deep Echo State Network (DeepESN)

(1) Echo State Network (ESN)

The traditional RNNs, such as LSTM and GRU, are trained by backpropagation (BP) of the error through the network. Due to the chain rule of BP, RNN training has the tendency to explode or vanish, requiring extensive computational resources to convergence, and can lead to poor local minima. Then the echo state network (ESN) proposes another trend toward training and employing RNNs. The core part of ESN is a single reservoir consisting of a large number of neurons that are randomly interconnected and/or self-connected. The architecture of a generic ESN model is shown in Figure 3, it consists of an input layer, a recurrent layer called reservoir, and an output layer. The input connections to the reservoir are randomly assigned and not trainable. The reservoir represents the hidden layer in the network with random and fixed recurrent connections. The only weights that need to be trained are the weights between the reservoir and the output layer, and the weights are trained through linear regression instead of

BP. Because of these characteristics, ESN becomes an efficient and powerful approach to RNN training, especially suited for time series modeling.

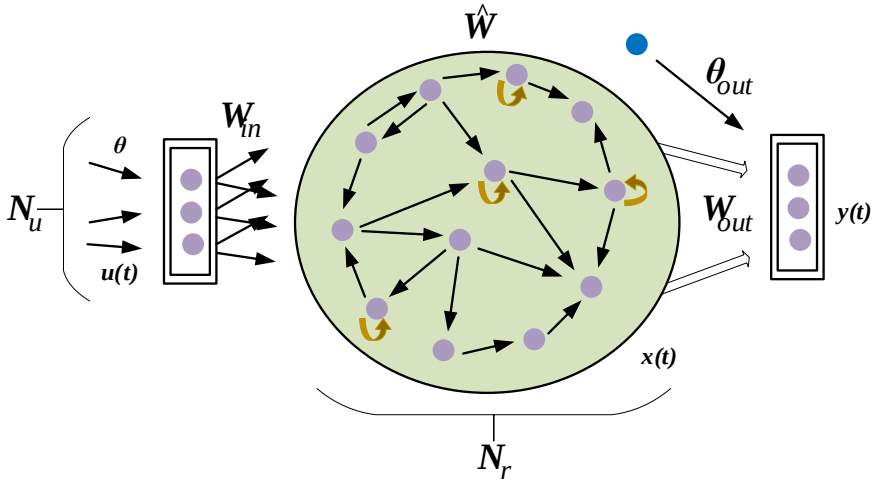


FIGURE 3 A GENERIC ESN MODEL.

Following the notations and terminologies introduced by Gallicchio et al. (2017), an ESN is a recurrent network model that contains N_u input units and N_r reservoir units. Omitting the bias terms for the ease of notation, the state of the reservoir at time t is updated according to the following recurrence function:

$$x(t) = (1 - a) \cdot x(t-1) + a \cdot \tanh(W_{in} \cdot u(t) + \widehat{W} \cdot x(t-1))$$

Where $x(t)$ represents the reservoir state at time t, $u(t)$ is the input state at time t, W_{in} denotes the input-to-reservoir weight matrix, $\widehat{W}$ expresses the recurrent reservoir weight matrix, $a \in [0,1]$ is the leaking rate, and tanh represents the element-wise application of the hyperbolic tangent activation function. The reservoir parameters are assigned randomly and then left untrained. W_{in} is randomly selected from a uniform distribution and then rescaled to have a maximum singular value equal to an input scaling parameter. $\widehat{W}$ is also randomly assigned from a uniform distribution and then rescaled. In conclusion, the current reservoir state depends on a linear combination of the previous reservoir state and the activation of the weighted current input state and weighted previous reservoir state.

The output of the ESN network is computed as a linear combination of the activated reservoir states, as follows:

$$y(t) = W_{out} \cdot x(t)$$

Where $y(t)$ denotes the output value at time t, W_{out} stands for the reservoir-to-output weight matrix. W_{out} is the only part that gets trained, typically using pseudo-inversion or ridge regression methods (Lukosevicius and Jaeger 2009).

(2) Deep Echo State Network

By stacking multiple reservoir layers on top of each other, Gallicchio et al. (2018) introduced an ESN-based deep learning architecture, called a deep echo state network (DeepESN). The architecture of a DeepESN is shown in Figure 4, it consists of an input layer, a stacked hierarchy of reservoirs, and an output layer. The input connections to the first reservoir are randomly assigned and not trainable, and each successive reservoir-to-reservoir layer is fed by the output of the previous one in the stack. The reservoir represents the hidden layer in the network with random and fixed recurrent connections. The only weights that need to be trained are the weights between the reservoirs and the output layer. For the sake of simplicity, we consider a hierarchical reservoir setup with N_L recurrent layers each of which contains the same number of units N_R.

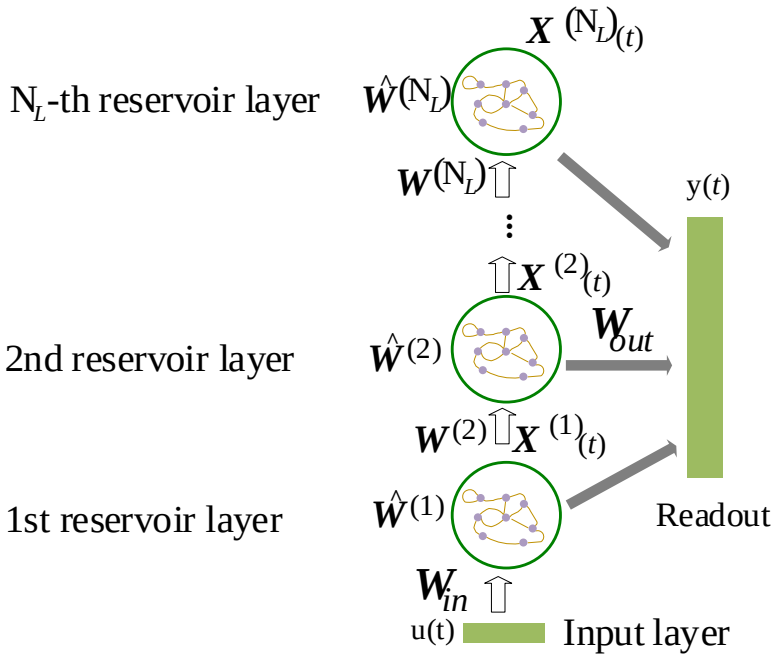


FIGURE 4 ILLUSTRATION OF A DEEP ECHO STATE NETWORK.

Omitting the bias terms for the ease of notation, the state of the first reservoir at time t is updated according to the following function:

$$x^{(1)}(t) = \left(1 - a^{(1)}\right) \cdot x^{(1)}(t-1) + a^{(1)} \cdot \tanh(W_{in} \cdot u(t) + \hat{W}^{(1)} \cdot x^{(1)}(t-1))$$

While for every layer $l > 1$, the state of the lth reservoir at time t is updated according to the following recurrence function:

$$x^{(l)}(t) = \left(1 - a^{(l)}\right) \cdot x^{(l)}(t-1) + a^{(l)} \cdot \tanh(W^{(l)} \cdot x^{(l-1)}(t) + \hat{W}^{(l)} \cdot x^{(l)}(t-1))$$

Where $u(t)$ is the input state at time t, l is the number of the reservoir layer ($l = 1,2, \dots, N_L$), $x^{(l)}(t)$ denotes the state of reservoir layer l at time t, W_{in} refers to the input weight matrix, $W^{(l)}$ is the inter-layer connection weight matrix from layer $(l-1)$ to layer l, $\hat{W}^{(l)}$ is the

recurrent weight matrix of reservoir layer l, $a^{(l)} \in [0,1]$ represents the leaking rate at layer l, and tanh is the element-wise application of the hyperbolic tangent. The reservoir parameters are assigned randomly and then left untrained. The input weight matrix W_{in} is randomly initialized from a uniform distribution and then rescaled to have a maximum singular value equal to an input scaling parameter w_{in}. The inter-layer weight matrix $W^{(l)}$ is also randomly initialized from a uniform distribution and then rescaled to have a maximum singular value equal to an inter scaling parameter w_{il}. The recurrent weight matrix $\widehat{W}^{(l)}$ is randomly initialized from a uniform distribution and then rescaled to have a spectral radius equal to ρ. In conclusion, the current reservoir state at layer l depends on a linear combination of the previous reservoir state at layer l and the activation of the weighted current reservoir state at layer $(l-1)$ and weighted previous reservoir state at layer l.

The output of the DeepESN is computed as a linear combination of the recurrent units over all of the recurrent layers, as follows:

$$y(t) = W_{out} \cdot \left[x^{(1)}(t) \cdot x^{(2)}(t) \cdots x^{(N_L)}(t)\right]^T$$

Where W_{out} denotes the weight matrix between reservoirs to output, which connects the reservoir units in all layers to the units in the output. The training of W_{out} out is conducted with ridge regression.

1.3.5 Evaluation Metrics

In order to evaluate the prediction performance for different models, three commonly used metrics are adopted, that are, the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the root mean square error (RMSE). The formulas of MAE, MAPE and RMSE are expressed as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right|$$

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

Where, N is the number of samples used in evaluation, y_i is the observed traffic volume in time step i at different stations, and $\hat{y}_i$ represents the predicted volume. Furthermore, it should be noted that the unit of MAE and RMSE for data collected under different time intervals is the number of vehicles per time scales (5 min). In the experiments, aiming to further evaluate the long-term prediction performance of all models, one-step ahead prediction is estimated. The

one-step ahead prediction is to forecast the traffic volume in the next 5 min from current time point.

1.4 RESULT

1.4.1 Data Set

The experimental data used in this project are collected from Regional Intelligent Transportation Information System (RITIS). The dataset includes traffic volume, speed and occupancy detected from microwave sensors in freeway network of Atlanta city. The data collection time starts from January 1st to January 15th in 2019. The time scale for the raw data samples is 1 minute, that is, 21600 raw data samples were accumulated for each sensor. In this project, the raw dataset is aggregated into 5 minutes, and thus, there were 4320 observations. By discarding the last 12 observations, a total of 4308 data samples were used in this experimental study. One sensor is selected in the city of Atlanta, as seen in Figure 5. Sensor A, located at latitude 33.7427 and longitude -84.4981, utilizes microwaves to detect traffic data northward on Interstate I-285(Western), which is a perimeter highway encircling the city of Atlanta. I-285 is recognized as one of the busiest and most congested roads in Atlanta.

FIGURE 5 THE SELECTED SENSOR A IN THE CITY OF ATLANTA.

1.4.2 Forecasting Results

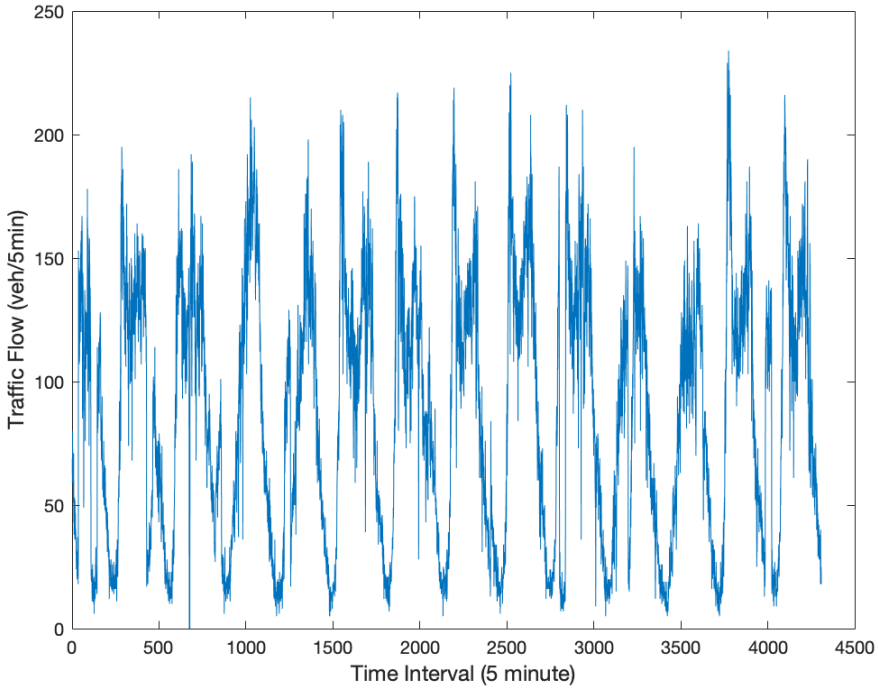


FIGURE 6 ORIGINAL TRAFFIC FLOW TIME SERIES OF SENSOR A.

Figure 6 shows the 4320 traffic volume series on I-285 detected from sensor A. From Figure 6, we can find that the original volume series exerts randomness and nonstationary in nature. Thus, to reduce the complexity of the volume series for better prediction, a second-decomposition method is proposed. Specifically, the CEEMDAN and WPD algorithms will be applied successively in the volume series decomposition.

(1) CEEMDAN Decomposition

The first step is to decompose the original volume series into multiple components through the CEEMDAN method. Following the CEEMDAN procedures outlined in the section 1.3.2, a total of 11 IMFs and one residual were obtained from the original volume data series, as seen in Figure 7. We can observe that the original volume series is decomposed into a series of relatively stationary data sets. The IMF with higher frequency represents the pattern of a shorter period, whereas the IMF with lower frequency represents the pattern of a longer period. Thus, the highly stochastic variations in the volume series are represented by the first few components (IMF1-IMF6), while the last few IMFs (IMF7-IMF11) describe the cyclical components in the volume series. The last component is the residual of the sifting process, which represents the overall trend of the volume series.

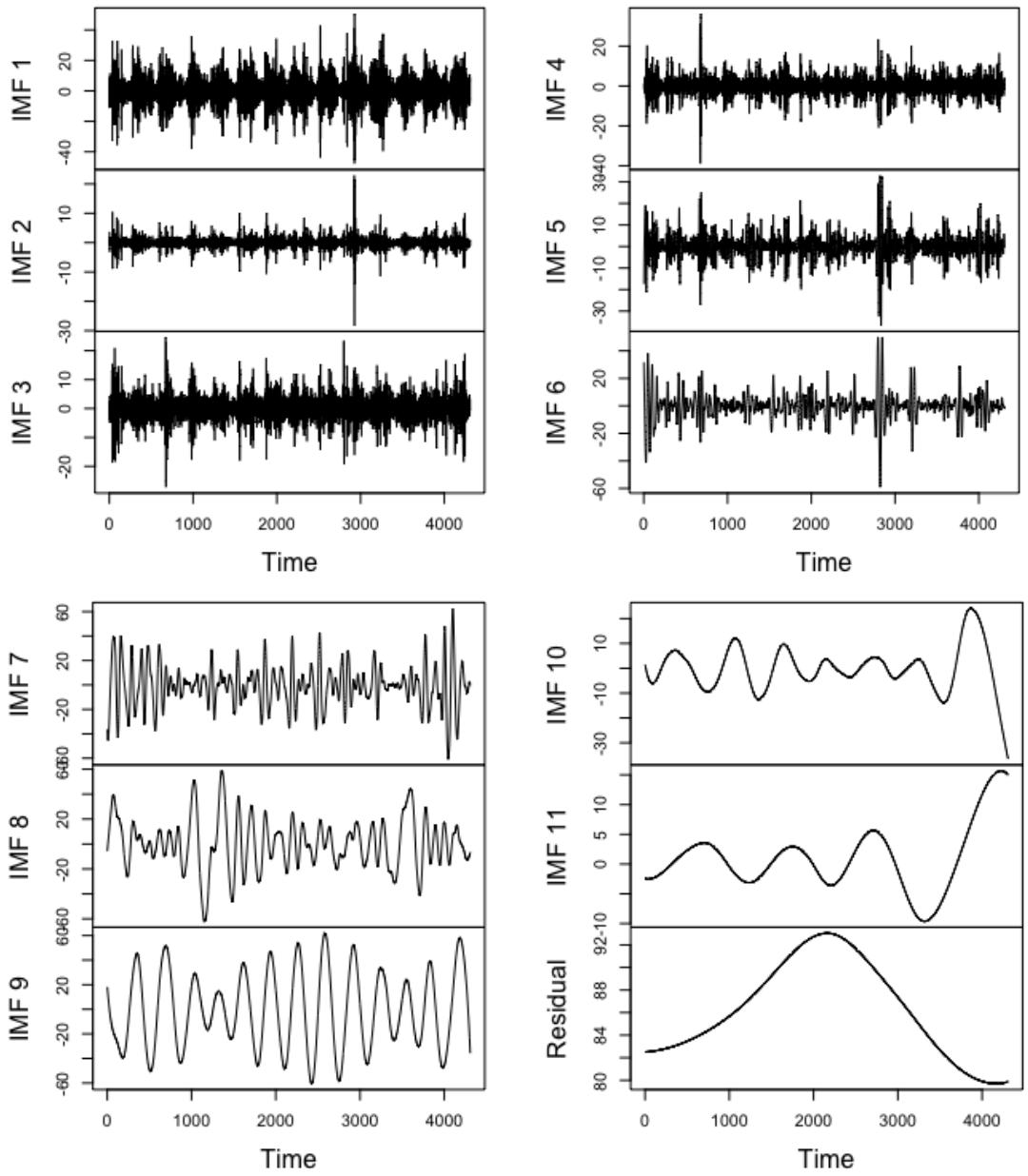


FIGURE 7 THE ORIGINAL TRAFFIC FLOW SERIES DECOMPOSED TO IMF1-IMF11 AND RESIDUAL BY THE CEEMDAN METHOD.

The trend persistence of decomposed components can be measured by the Hurst Exponent. If the Hurst value ranges between [0,0.5), it indicates a time series with long-term switching between high and low values in adjacent pairs, called anti-persistence in time series. If the Hurst value ranges between (0.5,1], it indicates a time series with long-term positive autocorrelation, called persistence in time series. When the Hurst Exponent value is equal to 0.5, it means the time series move in a random walk. The Hurst Exponent of each component decomposed by CEEMDAN is listed in Table 1. We can see that the Hurst Exponent values of IMF1, IMF2, IMF3, IMF4, IMF5 and IMF6 are between [0,0.5), indicating that these components

have anti-persistence features and will be further decomposed. The Hurst Exponent values of IMF7, IMF8, IMF9, IMF10, IMF11 and Residual are between (0.5,1], indicating that these components have persistence features.

TABLE 1 HURST EXPONENT OF EACH COMPONENT DECOMPOSED BY CEEMDAN.

Component	Hurst Exponent value	Component	Hurst exponent value
IMF1	0.29	IMF7	0.58
IMF2	0.27	IMF8	0.75
IMF3	0.13	IMF9	0.79
IMF4	0.16	IMF10	0.92
IMF5	0.32	IMF11	0.96
IMF6	0.45	Residual	1.00

(2) WPD Decomposition

Note that IMF1, IMF2, IMF3, IMF4, IMF5 and IMF6 obtained by CEEDMAN consist of the complex and high-frequency component signals of the volume series. So in the second step, WPD algorithm is used to further decompose these components, and the three-level WPD algorithm is adopted in this study. For example, Figure 8 shows the decomposition result of IMF1. Through the WPD method, IMF1 is further decomposed into 8 groups of wavelet packet coefficients, called IMF1_WPD1, IMF1_WPD2, IMF1_WPD3, IMF1_WPD4, IMF1_WPD5, IMF1_WPD6, IMF1_WPD7 and IMF1_WPD8. The same WPD decomposition procedure for IMF2 to IMF6. Therefore, the original volume series has been decomposed to a total of 54 components.

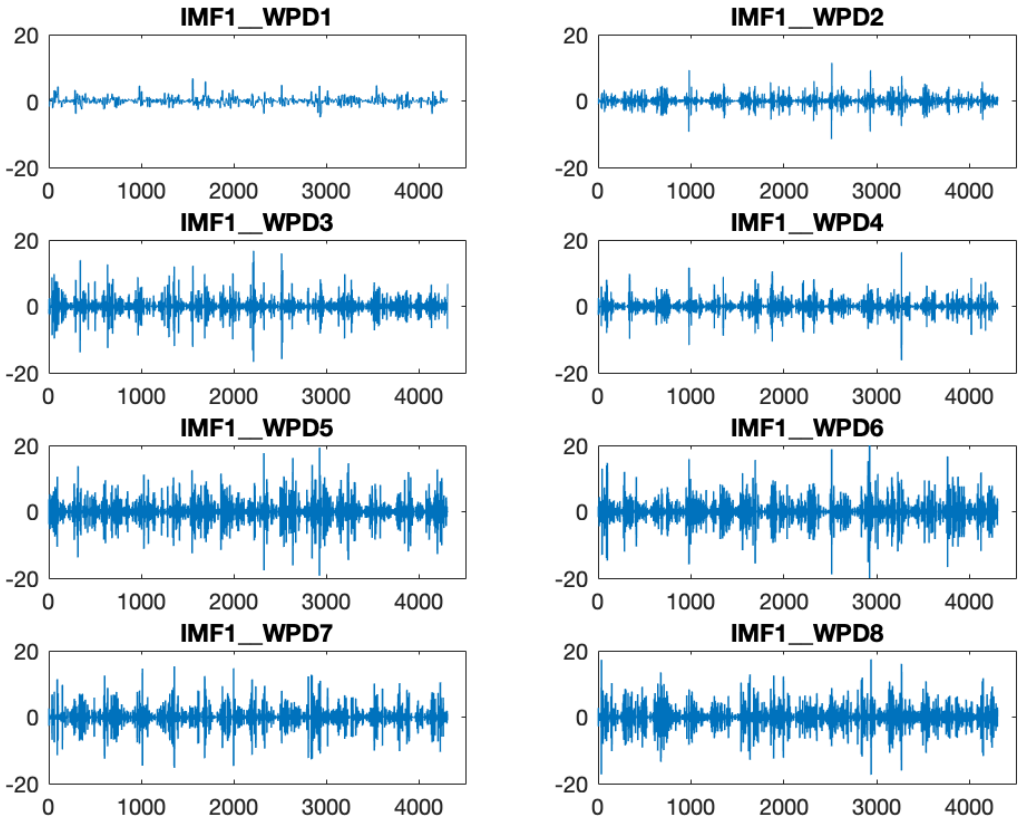


FIGURE 8 IMF1 DECOMPOSED TO IMF1_WPD1-IMF1_WPD8 COMPONENTS BY THE WPD METHOD.

(3) DeepESN Prediction

After the secondory decomposition procedure, the intrinsic characteristics of traffic volume series are decomposed to a batch of components, which should be more useful for prediction. In the third step, each of the 54 components obtained by the the CEEMDAN-WPD secondary decomposition method are divided into two subsets. The first 90% of data are used to train the forecasting model, and the last 10% are used to test the performance of the proposed forecasting model.

Following explanations in Section 1.3.4, we need to set up some parameters to train the DeepESN model. The pamameters include the number of reservoir units, the number of recurrent layers, spectral radius, input scaling, inter-layer scaling, leaking rate, and the readout regularization parameter of ridge regression. In addition, for simplification, all the layers share the same spectral radius ρ and the same inter-layer scaling w_{il}. Refering to the parameter configurations in Gallicchio et al. (2018) and Song et al. (2020), in this experiment, we choose the parameter selection set presented in Table 2 column 2, and the grid search technique is employed to select proper parameter values. The final parameter selection result is shown in Table 2 column 3. During the parameter selection process, we found that the prediction performance was more sensitive to the number of reservoir units, the number of recurrent

layers, spectral radius and leaking rate, while other parameters did not make a large difference in the prediction results.

TABLE 2 SELECTION OF HYPERMETERS USED IN DEEPESN MODEL.

Parameter	Selection Set	Final Selection
Number of reservoir units (N_r)	{10, 20, 50, 100}	20
Number of recurrent layers (N_l)	{3, 5, 7, 9, 15, 20}	5
Spectral radius (ρ)	{0.2, 0.4, 0.6, 0.8, 1.0}	0.8
Input scaling (w_{in})	{0.1, 0.2, 0.5, 1, 2, 5, 10}	0.5
Inter-layer scaling (w_{il})	{0.1, 0.2, 0.5, 1, 2, 5, 10}	0.5
Leaking rate (a)	{0.1, 0.3, 0.5, 0.7, 0.9}	0.9
Readout regularization (λ)	{0.1, 0.001, 1e-5, 1e-10}	1e-5

The DeepESN model is implemented on each of the 54 components, and adding the 54 predictions together formed the final prediction result. As shown in Figure 9, the observations and predictions of the test dataset fit almost perfectly, with MAPE of only 0.0007, illustrating the accuracy of the proposed method in this project. The exact numbers of errors are presented in Figure 10 and Table 3.

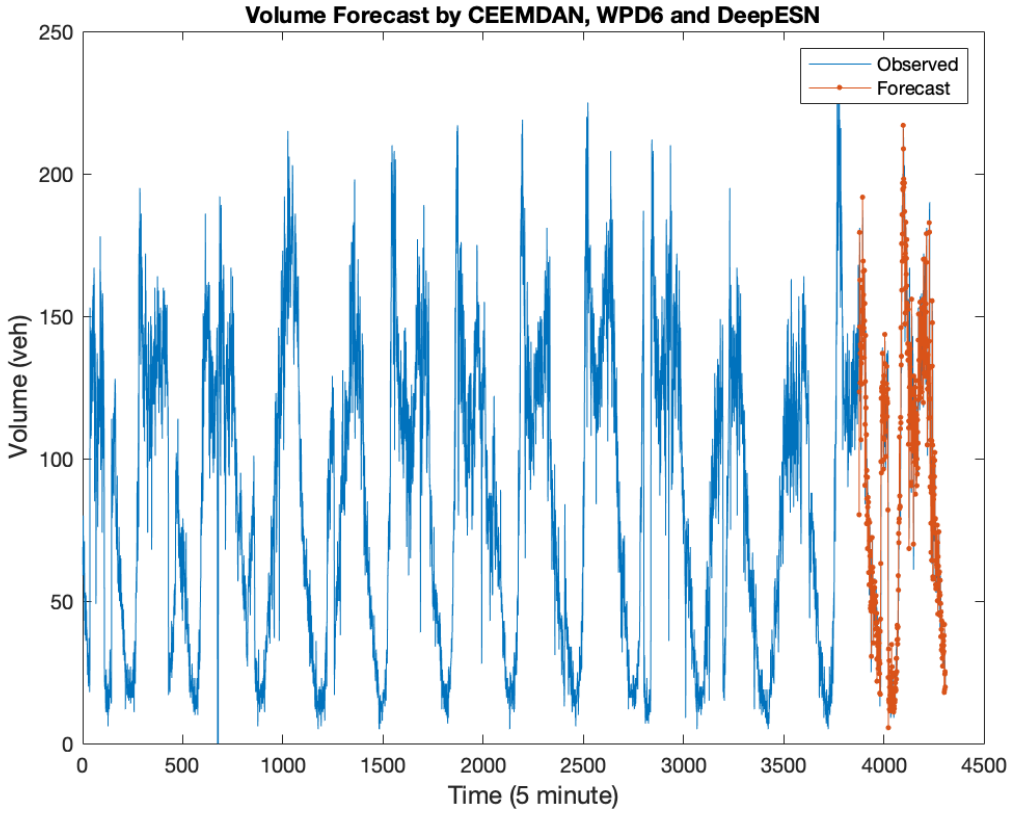
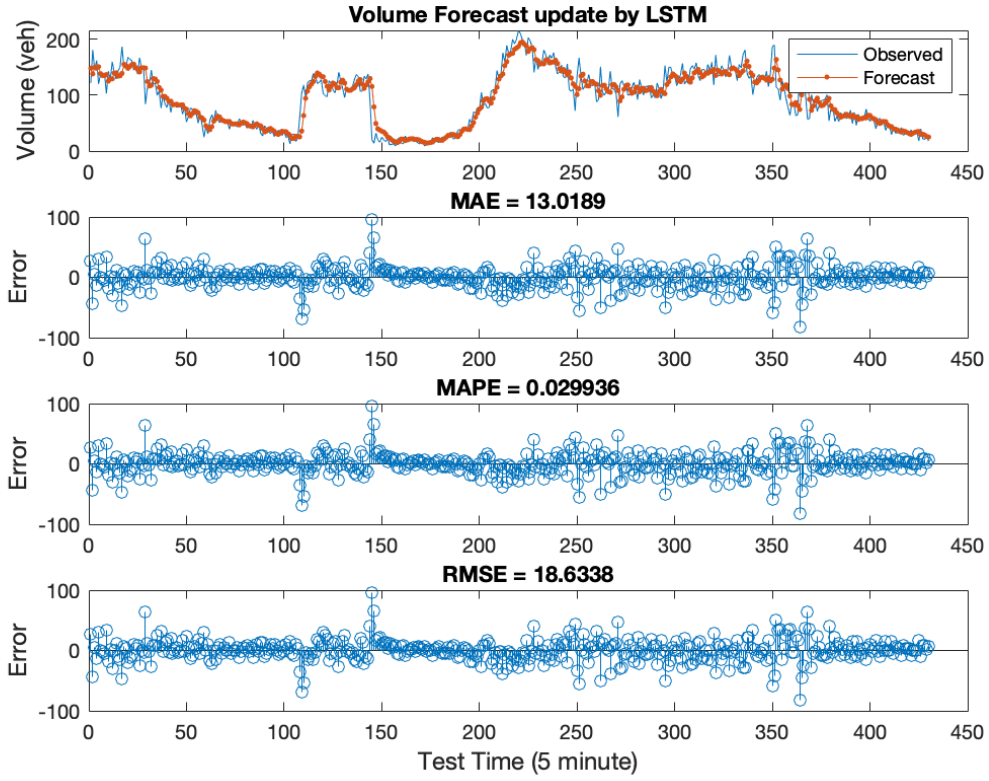


FIGURE 9 THE PREDICTION RESULT OF THE TEST DATA.

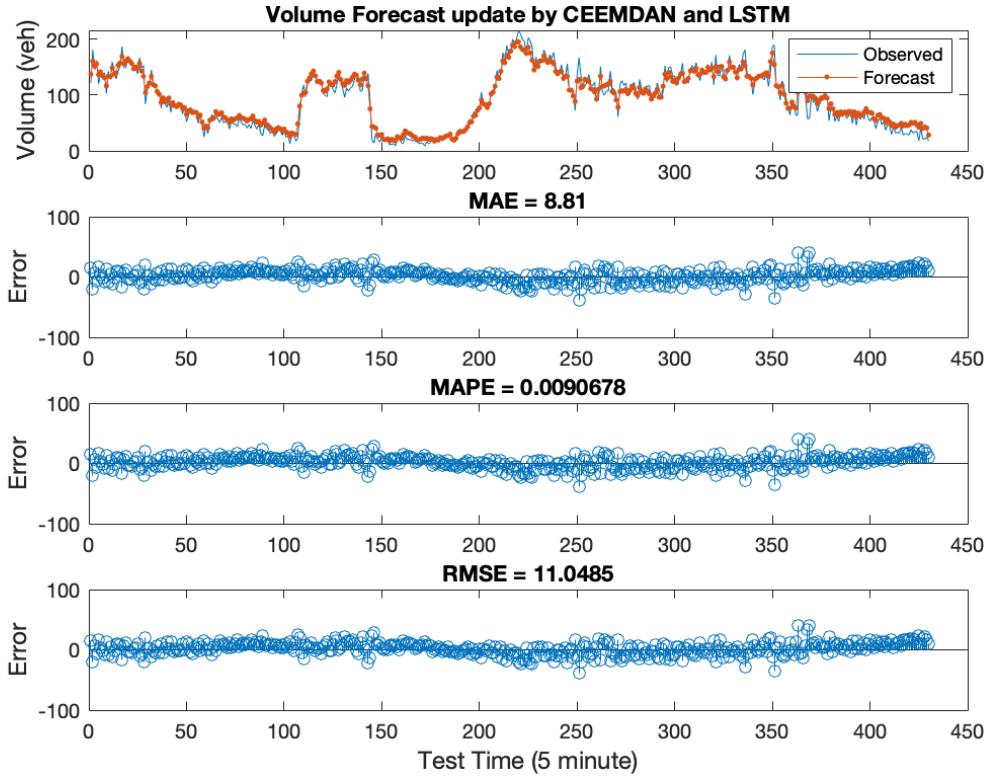
(4) Result Comparison with Base Models

In order to validate the effectiveness of the proposed model, seven predictive models are selected for comparative experiments on the same dataset. The seven predictive models include LSTM model, hybrid CEEMDAN-LSTM model, hybrid CEEMDAN-WPD-LSTM model, DeepESN model, hybrid CEEMDAN-DeepESN model, hybrid CEEMDAN-WPD-DeepESN model, and hybrid CEEMDAN-WPD6-DeepESN model. Specifically, the hybrid CEEMDAN-LSTM model means that the original traffic volume series is first decomposed to multiple IMFs, each of which is forecasted by the LSTM model, and then summing the predicted results of all the IMFs to obtain the final prediction result. In addition, the difference between CEEMDAN-WPD-DeepESN and CEEMDAN-WPD6-DeepESN is that in the secondary decomposition step, CEEMDAN-WPD-DeepESN utilizes WPD to decompose only the first IMF (IMF1), while CEEMDAN-WPD6-DeepESN utilizes WPD to decompose the first six IMFs (IMF1, IMF2, IMF3, IMF4, IMF5, IMF6).

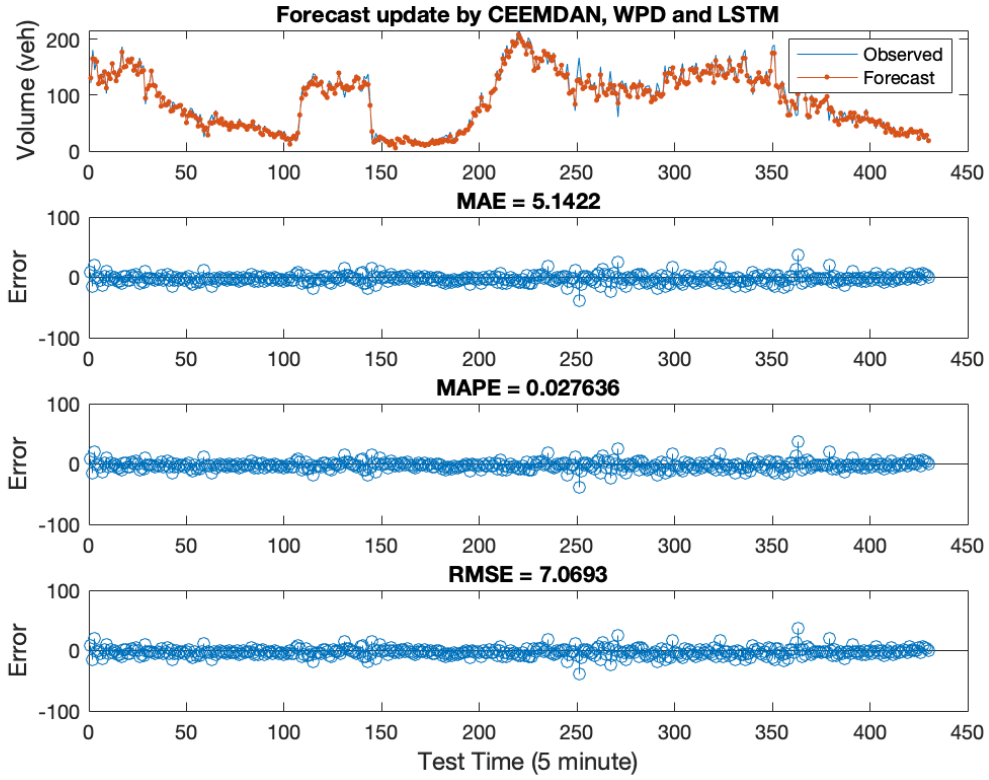
The comprehensive comparisons were implemented based on four evaluation criteria, that are MAE, MAPE, RMSE and elapsed time. The detailed prediction results of different models are presented in Figure 10, and the prediction evaluation results of different models are provided in Table 3.



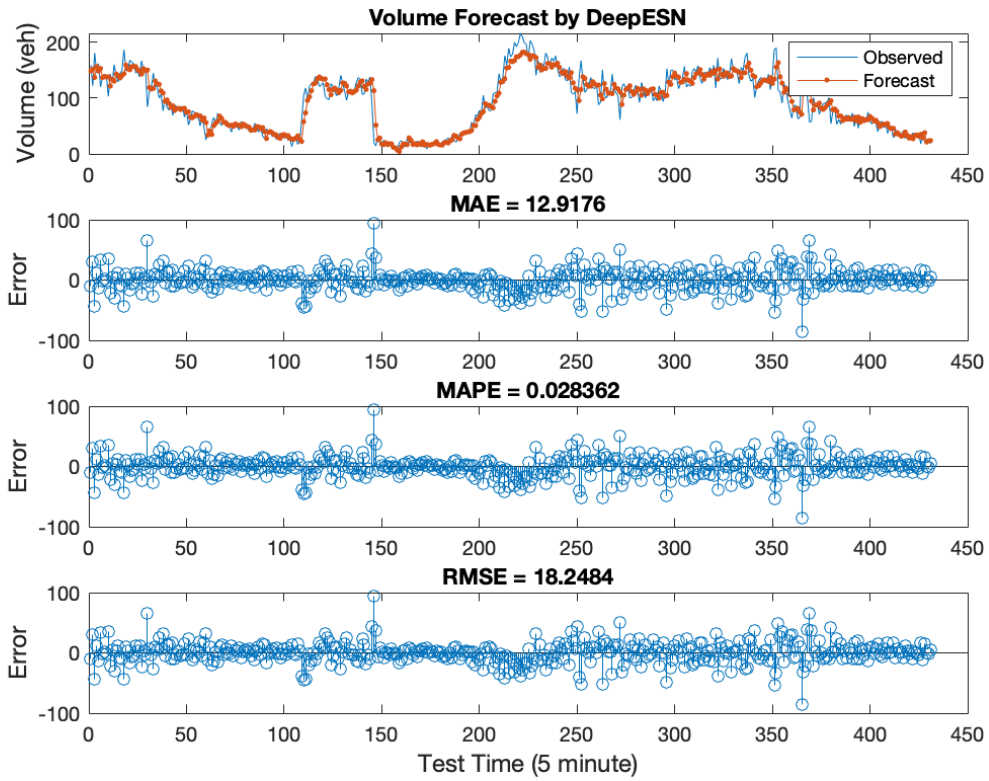
(a) Prediction result of the LSTM model.



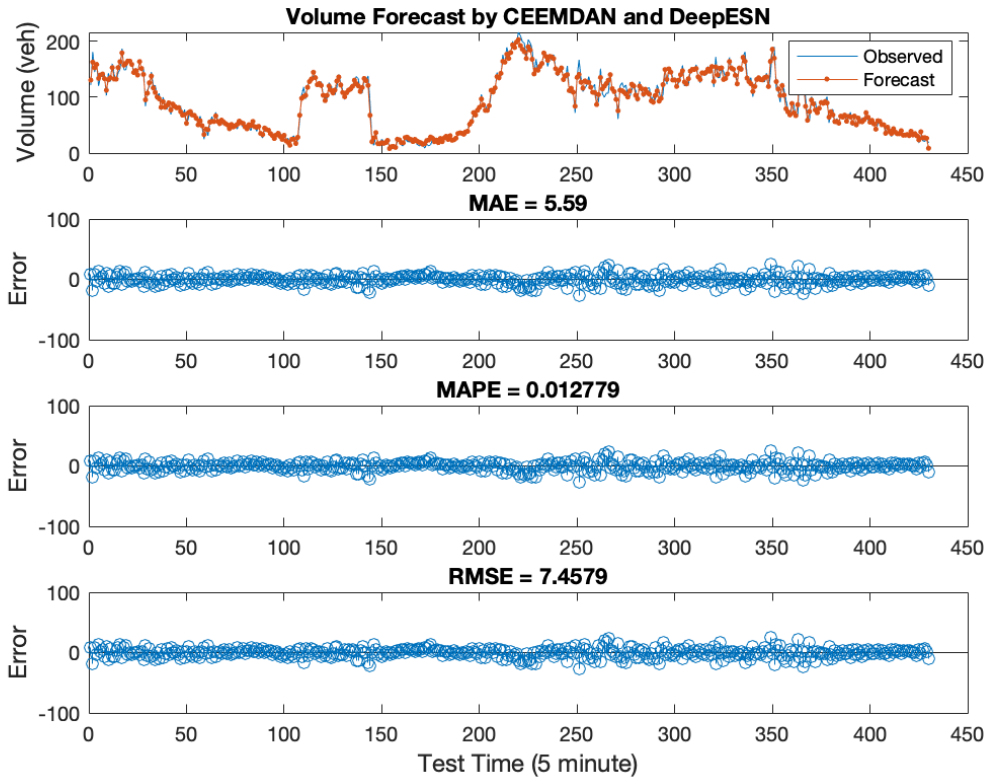
(b) Prediction result of the CEEMDAN-LSTM model.



(c) Prediction result of the CEEMDAN-WPD-LSTM model.

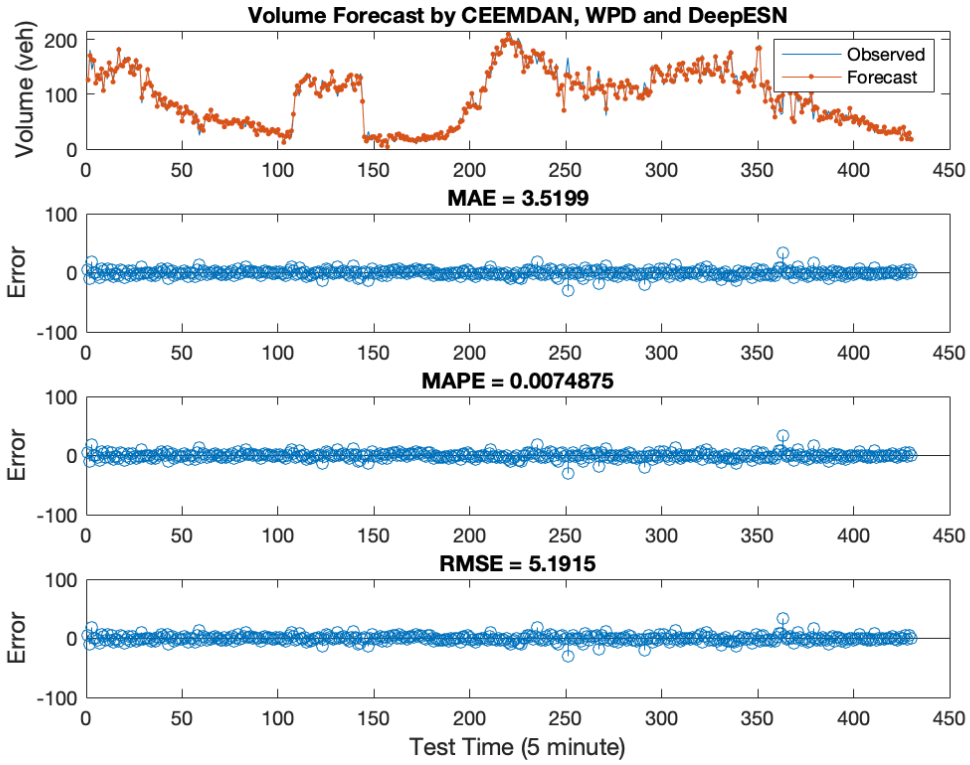


(d) Prediction result of the DeepESN model.

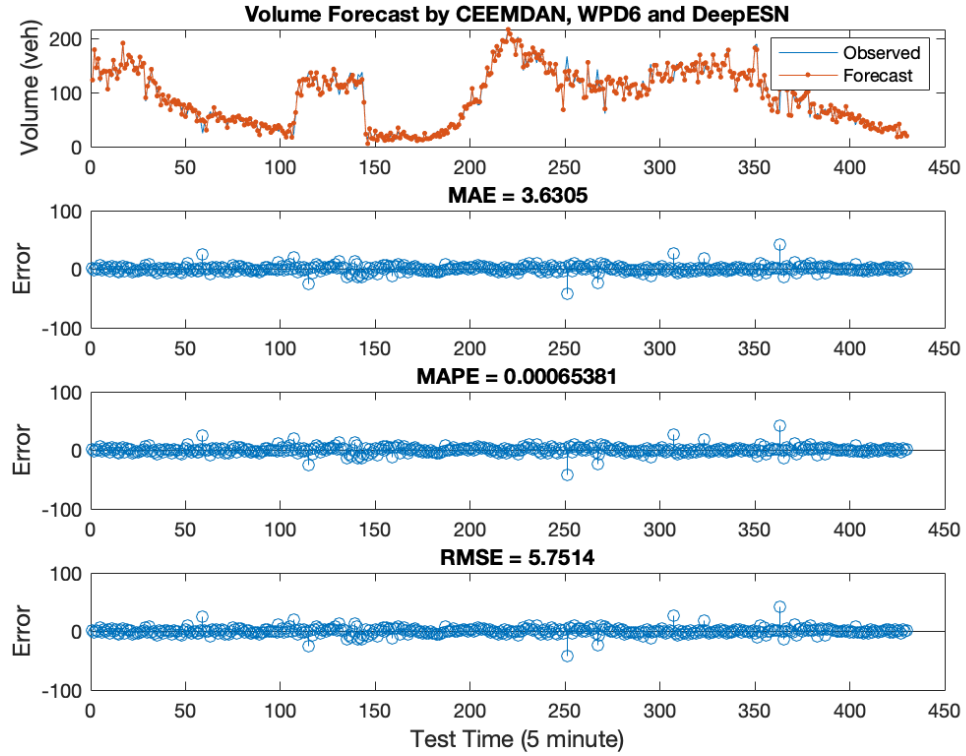


(e) Prediction result of the CEEMDAN-DeepESN model.

(f) Prediction result of the CEEMDAN-WPD-DeepESN model



(f) Prediction result of the CEEMDAN-WPD-DeepESN model.



(g) Prediction result of the CEEMDAN-WPD6-DeepESN model.

FIGURE 10 PREDICTION RESULTS OF DIFFERENT MODELS.

TABLE 3 PERFORMANCE EVALUATION RESULTS OF DIFFERENT MODELS.

Models	Computational Time(s)	MAE	MAPE	RMSE
LSTM	73	13.0	0.0299	18.6
CEEMDAN-LSTM	538	8.8	0.0091	11.0
CEEMDAN-WPD-LSTM	1309	5.1	0.0276	7.1
DeepESN	4	12.9	0.0284	18.2
CEEMDAN-DeepESN	5	5.6	0.0128	7.5
CEEMDAN-WPD-DeepESN	10	**3.5**	0.0075	**5.2**
CEEMDAN-WPD6-DeepESN	90	3.6	**0.0007**	5.8

From the observation of prediction results in Table 3 and Figure 10, several interesting finding can be summarized as follows:

(1) Both of the hybrid CEEMDAN-LSTM and CEEMDAN-WPD-LSTM models have smaller MAE, MAPE and RMSE values than those of the LSTM model, and all of the hybrid CEEMDAN-DeepESN, CEEMDAN-WPD-DeepESN and CEEMDAN-WPD6-DeepESN models have smaller MAE, MAPE and RMSE values than those of the DeepESN model. Smaller values for MAE, MAPE and RMSE indicate higher prediction accuracy. Therefore, we can conclude that the prediction results of models that incorporate decomposition techniques are superior to that without decomposition techniques. The main reason is that after the implementation of the data decomposition process, the high-frequency part in the original data are re-presented, and then the variation trend of the data is more obviously expressed. As a result, the model becomes more stable during training, resulting in higher prediction accuracy.

(2) The CEEMDAN-WPD-LSTM model has smaller MAE, MAPE and RMSE values than those of the CEEMDAN-LSTM model, and both of the CEEMDAN-WPD-DeepESN and CEEMDAN-WPD6-DeepESN models have smaller MAE, MAPE and RMSE values than those of the CEEMDAN-DeepESN model. The results demonstrate that the second-order decomposition technique has higher predictive accuracy than the first-order decomposition technique.

(3) The prediction accuracy of LSTM and DeepESN models is similar, while the CEEMDAN-DeepESN model has smaller MAE, MAPE and RMSE values than those of the CEEMDAN-LSTM model, and CEEMDAN-WPD-DeepESN model has smaller MAE, MAPE and RMSE values than those of the CEEMDAN-WPD-LSTM model. As a result, deep learning prediction methods combined with decomposition techniques to forecast traffic flow have proven feasible and effective. Moreover, the combination of DeepESN and decomposition techniques produces higher accuracy in predicting traffic flow than the combination of LSTM and decomposition techniques.

(4) The CEEMDAN-WPD-DeepESN and CEEMDAN-WPD6-DeepESN models have very similar MAE, MAPE and RMSE values. This phenomenon illustrates that in the second step of the

decomposition process, it is not that the more components to be decomposed, the better. Decomposing IMF1 alone not only yields the same prediction accuracy as decomposing IMF1, IMF2, IMF3, IMF4, IMF5 and IMF6, but also reduces the calculation time by a factor of nine.

(5) The DeepESN related models are much superior to the LSTM related models in calculating time consumption. For example, when predicting the same dataset, DeepESN only took 4 seconds while LSTM took 73 seconds, CEEMDAN-DeepESN only took 5 seconds while CEEMDAN-LSTM took 538 seconds, and CEEMDAN-WPD-DeepESN only took 10 seconds while CEEMDAN-WPD-LSTM took 1309 seconds. The DeepESN related models have proven to be powerfully efficient. The hybrid models of DeepESN and decomposition techniques have verified to be both accurate and efficient.

1.5 CONCLUSION

In this project, a novel hybrid prediction model, CEEMDAN-WPD-DeepESN, has been proposed for short-term traffic flow prediction. First, with consideration of the intrinsic characteristics of traffic flow series, the raw dataset is decomposed to IMFs by CEEMDAN. Then, the high frequency IMFs are decomposed to sub-series by WPD. Next, DeepESN is adopted to forecast each of the decomposed components. Finally, the forecasting results of the sub-series are aggregated to obtain the final prediction result. Experimental results show that, compared with the LSTM, CEEMDAN-LSTM, CEEMDAN-WPD-LSTM, DeepESN, and CEEMDAN-DeepESN models, the proposed CEEMDAN-WPD-DeepESN model has obvious advantages in improving the accuracy and efficiency of traffic flow prediction. Several findings are summarized. (1) The hybrid models have better prediction performance than the individual models. (2) The second-order decomposition technique CEEMDAN-WPD has higher predictive accuracy than the first-order decomposition technique CEEMDAN. (3) The DeepESN-related hybrid models produce higher accuracy than the LSTM-related hybrid models in traffic flow prediction. (4) It is not that the more components that are decomposed, the higher the prediction accuracy obtained, and the fast and good prediction results can be obtained by simply decomposing IMF1 in the second decomposition step. (5) DeepESN-related models has demonstrated strong abilities in efficient traffic flow forecasting.

2.0 TASK 2: SHORT-TERM TRAFFIC SPEED PREDICTION ON A ROAD NETWORK

2.1 INTRODUCTION

Traffic prediction is an important topic due to its potential in various intelligent applications. However, network-wide traffic state prediction is challenging due to the complicated spatial structure and temporal dependencies in road networks. Typically, traffic conditions at nearby locations affect each other. To capture such spatial structure feature in the road network, this study adopts the graph convolution technique. In addition, deep learning recurrent neural networks are employed to mine the temporal dependencies of traffic data.

2.1.1 Objective

This project compares several graph convolutional based deep learning models to forecast the network-wide traffic speed. The objectives are to: (1) Evaluate performance of graph convolutional methods in capturing the spatial dependency of the road network. (2) Evaluate the feasibility and superiority of the DeepESN model in predicting the traffic state of the road network. (3) Improve the accuracy and efficiency of short-term network-wide traffic speed forecasting.

2.1.2 Scope

The study focuses on the Atlanta Road network. The experiment studied 15 days of data from 01/01/2019 to 01/15/2019 with a total of 501 sensors.

2.2 LITERATURE REVIEW

Although deep learning models have shown their superior capability for traffic flow prediction in the above task, these methods ignore the spatial dependencies, so that the traffic state prediction is not constrained by the structure of the road network, which may lead to failure in forecasting the entire road network. Aiming at this problem, recent studies have focused on graph-structured data, which can be applied to capture the spatial relationships present in a traffic network. For example, Yu, Lee, and Sohn (2020) designed a graph-based traffic prediction learning model, which simulates real traffic propagation based on a variable adjacency matrix. Lv et al. (2021) used graphs to encode non-Euclidean spatial and semantic correlations between roads and captured the correlations in conjunction with graph convolutional networks. Cui, Henrickson, and Ke (2020) treated the road network as a graph and proposed a Graph Convolutional Long Short-Term Memory Neural Network (GC-LSTM) to learn the spatiotemporal connections between roads for further prediction. Zhao et al. (2018) proposed a traffic forecasting method T-GCN that combines the graph convolutional network (GCN) with the gated recurrent unit (GRU) to capture the spatial and temporal dependences.

Therefore, in this study, we also learn the traffic network as a graph and conduct convolution on the traffic network-based graph. Different deep learning models combined with the graph convolution are applied to the network-wide traffic state prediction, to observe the effectiveness and limitations of the graph convolutional deep learning models.

2.3 METHODOLOGY

2.3.1 Graph Convolutional Long Short Term Memory (GC-LSTM)

The architecture of the Graph Convolutional LSTM (GC-LSTM) recurrent neural network is presented in Figure 11, which learns both the spatial dependencies and the dynamic temporal dependencies in traffic series. In this model, the gates structure in the LSTM and the hidden state are unchanged, but the cell state from the previous time step is incorporated by the graph convolution features. The forget gate f_t, the input gate i_t, the output gate o_t, and the input cell state $\tilde{C}_t$ in terms of time step t are defined as follows.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Where $\cdot$ denotes the matrix multiplication operator. W_f, W_i, W_o, and W_C are the weight matrices, mapping the input to the three gates and the input cell state. b_f, b_i, b_o, and b_C represent bias vectors. σ is the gate activation function, and $\tanh$ is the hyperbolic tangent function.

Due to each node in a traffic network graph is influenced by the preceding states of itself and its neighboring nodes, the LSTM cell state of each node in the graph should also be affected by neighboring cell states. Thus, a cell state gate is designed and added in the LSTM cell. The cell state gate, as shown in Figure 11, is defined as follows.

$$C^*_{t-1} = W * A \cdot C_{t-1}$$

Where W is a weight matrix to measure the contributions of neighboring cell states. To correctly reflect the traffic network structure, W is constrained by multiplying an adjacency matrix A. With this gate, the influence of neighboring cell states will be considered when the cell state is recurrently input to the subsequent time step. Then, the final cell state and the hidden state are calculated as follows.

$$C_t = f_t * C^*_{t-1} + i_t * \tilde{C}_t$$

$$h_t = o_t * \tanh(C_t)$$

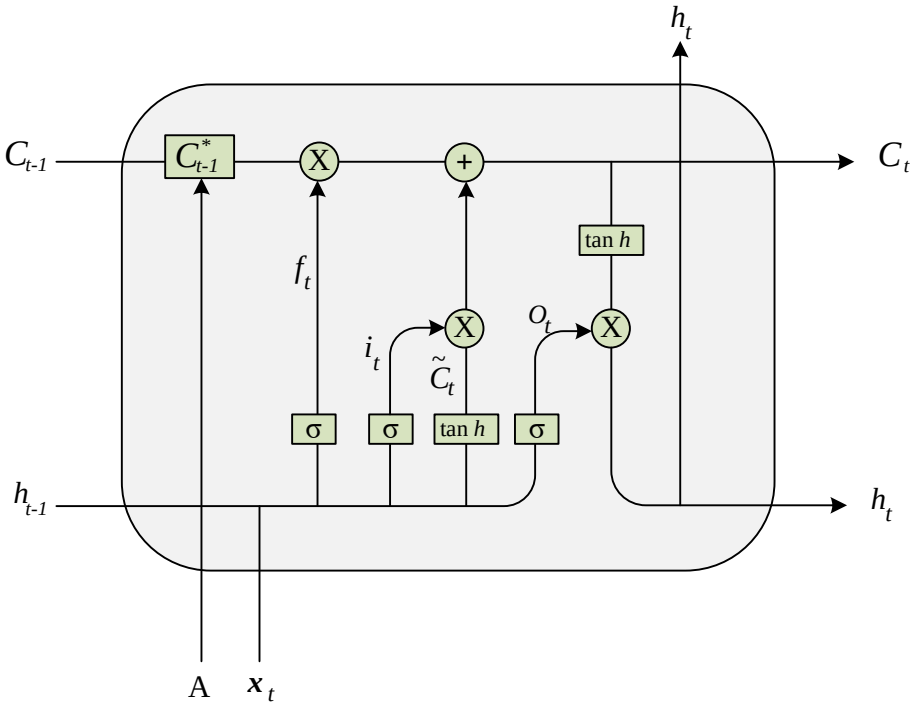


FIGURE 11 THE ARCHITECTURE OF THE GC-LSTM MODEL.

2.3.2 Graph Convolutional Neural Network with Gated Recurrent Unit (GCN-GRU)

GCN was proposed by Kipf and Welling (2016), and its origin is the spectral-graph CNN introduced by Bruna et al. (2013). In general, GCN generalizes convolutional operations to non-Euclidean domains based on the spectral graph theory. The working principle of GCN can be briefly described as follows.

Given a graph $G = (V, E)$, V represents the set of vertices, and E is the set of edges. A GCN requires two matrices as input, they are the feature matrix X and the adjacency matrix A. The hidden layer of a GCN is denoted as $H^l = f(H^{l-1}, A)$ where the initial hidden layer H^0 is set to X and f indicates the propagation rule. The simplest form of a layer propagation rule is: $H^l = \sigma(AH^{l-1}W^{l-1})$, where σ represents an activation function, and W^{l-1} is a weight parameter matrix. However, the simplest layer propagation rule has two typical problems. First, only the features of neighboring nodes can be transferred to the next layer, while the node's own features cannot be passed to the next layer. To address this limitation, a new adjacency matrix $\tilde{A}$ is created by adding an identity matrix I to the given adjacency matrix A. Second, feature values are amplified as they propagate layer by layer. To circumvent this problem, the spectral normalization ($D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}$) serves as a feasible solution. Therefore, the final propagation rule of GCN established by Kipf and Welling (2016) takes the form of the following equation.

$$H^l = \sigma(D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}H^{l-1}W^{l-1})$$

This study will combine GCN and GRU models to capture the spatial and temporal features from the road network data. Figure 12 illustrates the architecture of the GCN-GRU model, where h_{t-1} is the old hidden state output at time $(t-1)$, z_t and r_t denote the update and reset gates at time t, and h_t is the new hidden state output at time t.

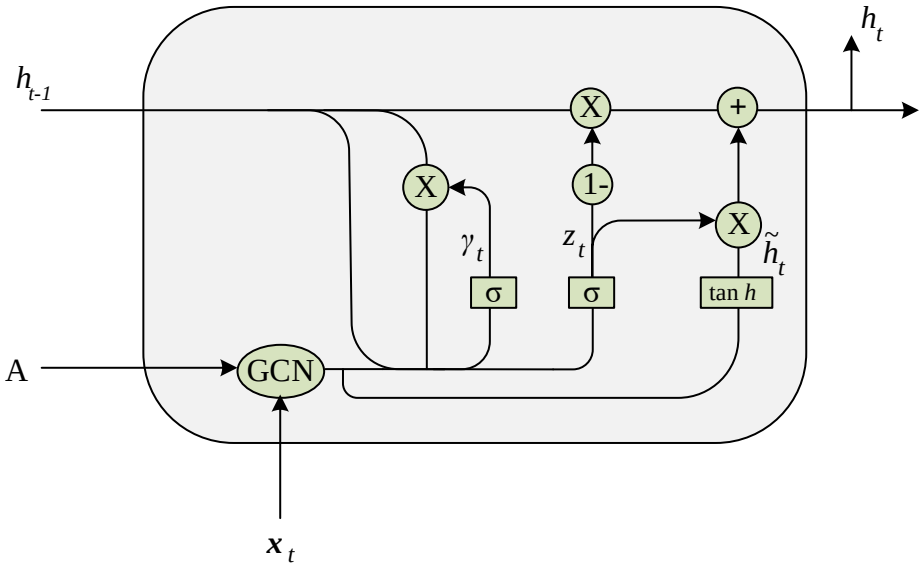


FIGURE 12 THE ARCHITECTURE OF THE GCN-GRU MODEL.

Each of the parameters in the GCN-GRU model is calculated as follows.

$$z_t = \sigma(W_z[h_{t-1}, f(A, x_t)] + b_z)$$

$$r_t = \sigma(W_r[h_{t-1}, f(A, x_t)] + b_r)$$

$$\tilde{h}_t = \tanh(W_h[r_t * h_{t-1}, f(A, x_t)] + b_h)$$

$$h_t = z_t * \tilde{h}_t + (1 - z_t) * h_{t-1}$$

Where f represents the process of GCN, W_z, W_r, and W_h are the weight matrices, mapping the input to the two gates and the input cell state. b_z, b_r, and b_h are bias vectors. σ is the gate activation function, and $\tanh$ is the hyperbolic tangent function. The equations can be used to capture the spatiotemporal dependencies of the road network, and then realize the network-wide traffic state prediction based on the extracted spatiotemporal features.

2.3.3 Loss Function

During the training process of the GC-LSTM and GCN-GRU models, there will be some error between the actual value y_t and predicted output $\hat{y}_t$. To minimize the error, a loss function needs to be defined, and then an algorithm used to optimize the weights, thereby optimizing the model. In this study, mean squared error (MSE) is adopted as the loss function, as shown below.

$$L = \frac{1}{N \cdot T} \sum_{t=1}^{T} (\hat{y}_t - y_t)^2$$

2.4 RESULT

2.4.1 Data Set

The experimental data used in this study mainly consists of two parts. The first part is the traffic speed series, which were collected from Regional Intelligent Transportation Information System (RITIS). The data collection time period starts from January 1st to January 15th in 2019. The time sale for the raw data samples is 1 minute, that is, 21600 raw data samples were accumulated for each sensor. In this project, the raw dataset is aggregated into 5 minutes, and thus, there are 4320 observations for each sensor. By discarding the last 11 observations, 4309 data samples of each sensor are remained for study. Moreover, after discarding some sensors that have many empty values, a total of 501 sensors in the city of Atlanta are selected for this experimental study, as seen in Figure 13. The data is normalized via the min-max normalization method. We use 70% of the data for training, and 30% for testing.

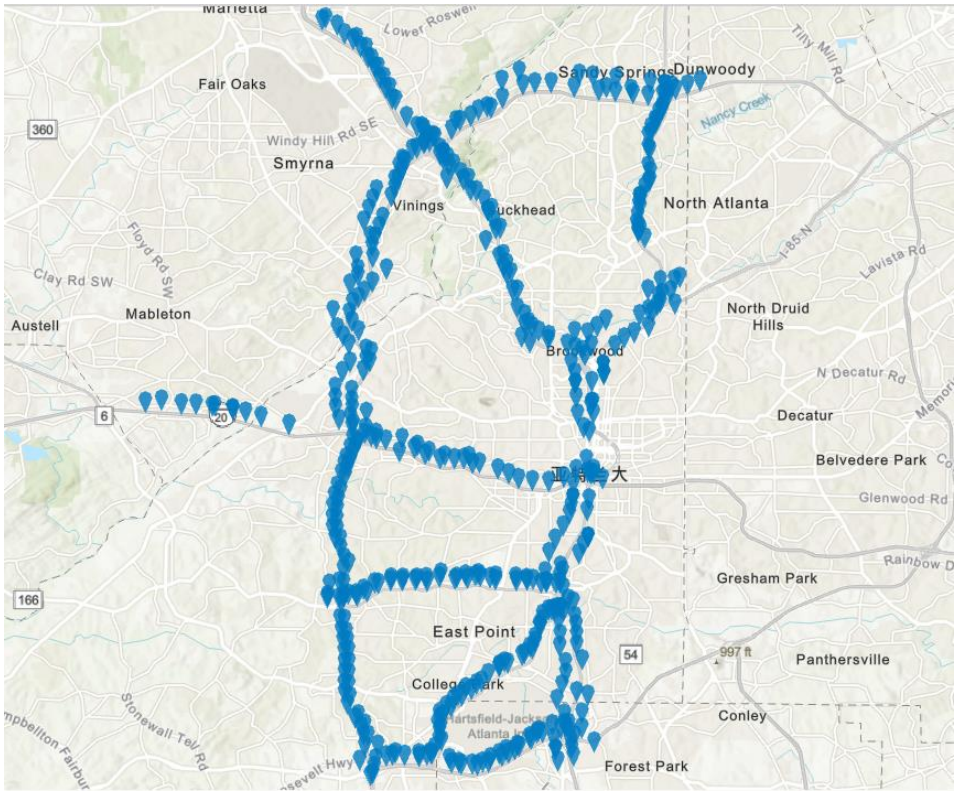


FIGURE 13 THE SELECTED 501 SENSORS IN THE CITY OF ATLANTA.

The second part of the data is the adjacency matrix of the 501 sensors. In this study, the widely used distance matrix is adopted to represent the spatial closeness between sensors. The distance-based adjacency matrix is defined as:

$$A_{ij} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right), if\ \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq \epsilon, i \neq j \\ 1, if\ i = j \\ 0, otherwise \end{cases}$$

Where d_{ij} is the road network distance from sensor i to sensor j, σ is the standard deviation (set as 0.1 in this experiment), and ϵ (set as 0.001 in this experiment) is the threshold to control the sparsity of the adjacency matrix A. A_{ij} depicts the spatial relationship between sensor i and sensor j. $A_{ij} = 0$ indicates speed data of sensor i and sensor j are uncorrelated, while $A_{ij} = 1$ indicates sensor i and sensor j are neighborhoods and the speed of sensor i is related to the previous speeds of itself and its neighbors, including sensor j.

2.4.2 Simulation Environment

The environment configuration of the simulation experiment is as follows: (1) Hardware platform. Processor: 2.4GHz Quad-Core Intel Core i5; Memory: 16GB 2133MHz LPDDR3; Graphics: Intel Iris Plus Graphics 655 1536MB; Operating system: 64-bit operating system. (2) Software environment. Python 3.9.4 and PyTorch 1.7.1 deep learning frameworks are used to simulate the GC-LSTM, GCN-GRU and DeepESN models.

2.4.3 Model Parameter Selection

The parameters of the GC-LSTM model are learning rate, training epoch batch, size and number of hidden layers. In this experiment, the learning rate was set at 0.001, the batch size was 20, the training epoch was 3, and the number of hidden layers was set as 20.

The parameters of GCN-GRU model are learning rate, training epoch batch, size and number of hidden layers. In this experiment, the learning rate was set at 0.001, the batch size was 20, the training epoch was 3, and the number of hidden layers was set as 20. The activation function of the first layer is a rectified linear unit (ReLU), and the activation function of the second layer is Softmax.

The parameters of DeepESN model include the number of reservoir units, the number of recurrent layers, spectral radius, input scaling, inter-layer scaling, leaking rate, and the readout regularization parameter of ridge regression. In this experiment, the number of reservoir units is set as 5, the number of recurrent layers is 2, the spectral radius is 0.8, the input scaling is 0.3, the inter-layer scaling is 0.3, the leaking rate is 0.9, and the readout regularization parameter is 0.0001.

2.4.4 Forecasting Results

TABLE 4 COMPARISON RESULTS OF DIFFERENT MODELS.

Models	MSE	Computational Time(s)
LSTM	0.0313	10.0
DeepESN	**0.0045**	37.8
GC-LSTM	0.0056	10.5

GCN-GRU	0.0058	882.1
GC-DeepESN	0.0138	52.6

In this experiment, five predictive models are selected on the same dataset and adjacency matrix. The five predictive models include the LSTM model, the DeepESN model, the GC-LSTM model, the GCN-GRU model, and the GC-DeepESN model. The comprehensive comparisons are implemented based on the evaluation index MSE. The prediction evaluation results of different models are provided in Table 4. From Table 4, several interesting findings can be summarized as follows.

(1) The MSE produced by the GC-LSTM model is smaller than that of the individual LSTM model. Therefore, we can say that considering the spatial dependence of the road network is beneficial for traffic speed forecasting, which can improve the accuracy of prediction.

(2) The MSE produced by the individual DeepESN model is smallest, even smaller than that of the GC-LSTM and GCN-GRU models. This phenomenon once again proves the superiority of the DeepESN method in predicting traffic states, both for an individual sensor and for the entire road network.

(3) A strange phenomenon is that when the DeepESN model takes into account the spatial dependence of the road network, the prediction effect becomes worse. This shows that for different forecasting models, it is necessary to consider the connection matrix of the road network with caution. For different models, the selection of connection matrix needs to be studied more systematically.

2.5 CONCLUSION

In this task, the prediction of traffic state is extended to the network wide. In order to improve the prediction accuracy of the road network, the deep learning models are combined with the graph convolutional networks to capture the spatial-temporal characteristics of the traffic state in the road network. The traffic speed data collected from 501 sensors in the city of Atlanta are used for the experimental study. Five predictive models, including the LSTM model, the DeepESN model, the GC-LSTM model, the GCN-GRU model and the GC-DeepESN model, are implemented in the traffic speed data. The experimental results reveal the superiority of the DeepESN model in predicting the traffic speed of the road network. In addition, the LSTM model combined with the graph convolutional network achieves better prediction performance than the individual LSTM model. However, the DeepESN model combined with the graph convolutional neural network yields worse prediction performance than the individual DeepESN model.

3.0 RECOMMENDATIONS

This study can be extended in a number of directions. For task 1, the secondary decomposing components needs to be selected more systematically and accurately in the future. For task 2, DeepESN is a relatively new model, fully exploiting its potential especially in combination with

graph convolution methods is important. The success in this regard will bring large-scale network-level spatiotemporal prediction a step forward.

In addition, the data used in this report are from RITIS, located in the city of Atlanta. In the future, we can apply the methodology proposed in this report to other datasets. There are some open data resources, for example, METR-LA dataset containing traffic speed and volume collected from the highway of the Los Angeles County Road network, with 207 loop detectors, and Performance Measurement System (PeMS) dataset containing over 18,000 sensors (e.g., inductive loops, side-fire radar, magnetometers) among the metropolitan areas of California.

Real-time traffic state prediction is of great importance in ITS, because accurately forecasting traffic states can help drivers better plan their routes, shorten travel time and improve traffic conditions, and thus alleviate the issues of traffic congestion. The methodologies proposed in this report can be applied to navigation map companies. When the driver navigates through the map APP, based on the map's function of predicting the future traffic state in real time, it will recommend the fastest route for the driver to avoid the congested road.

4.0 REFERENCE LIST

[1] Okutani, I., and Y. J. Stephanedes. "Dynamic prediction of traffic volume through Kalman filtering theory." Transportation Research Part B Methodological 18.1(1984):1-11.

[2] Whittaker, J., S. Garside, and K. Lindveld. "Tracking and predicting a network traffic process." International Journal of Forecasting 13.1(1997):51-61.

[3] Stathopoulos, A., and M. G. Karlaftis. "A multivariate state space approach for urban traffic flow modeling and prediction." Transportation Research Part C: Emerging Technologies 11.2(2003):121-135.

[4] Ma, M., et al. "Short-term traffic flow prediction using a self-adaptive two-dimensional forecasting method." Advances in Mechanical Engineering 9.8(2017).

[5] Ahmed, M. S., and A. R. Cook. "Analysis of freeway traffic time-series data by using Box-Jenkins techniques." Transportation Research Record Journal of the Transportation Research Board 722(1979):1-9.

[6] Levin, M., and Y. D. Tsao. "On forecasting freeway occupancies and volumes." Transportation Research Record Journal of the Transportation Research Board 773(1980):47-49.

[7] Hamed, Mohammad M., H. R. Al-Masaeid, and Z. M. B. Said. "Short-Term Prediction of Traffic Volume in Urban Arterials." Journal of Transportation Engineering 121.3(1995):249-254.

[8] Voort, Mvd, M. Dougherty, and S. Watson. "Combining Kohonen maps with ARIMA time series models to forecast traffic flow." Transportation Research Part C Emerging Technologies 4.5(1996):307-318.

[9] Lee, S., et al. "Application of Subset Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting." Transportation Research Record Journal of the Transportation Research Board 1678.1(1999):179-188.

[10] Williams, B. M., P. K. Durvasula, and D. E. Brown. "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models." Transportation Research Record Journal of the Transportation Research Board 1644(1998):132-141.

[11] Williams, B. M. "Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling." Transportation Research Record Journal of the Transportation Research Board 1776.1(2001):194-200.

[12] Williams, B. M., and L. A. Hoel. "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results." Journal of Transportation Engineering 129.6(2003):664-672.

[13] Chandra S. R., and H. Al-Deek. "Predictions of freeway traffic speeds and volumes using vector autoregressive models." Journal of Intelligent Transportation Systems Technology, Planning, and Operations 13.2(2009):53-72.

[14] Nicholson, H., and C. D. Swann. "The prediction of traffic flow volumes based on spectral analysis." Transportation Research 8.6(1974):533-538.

[15] Tchrakian, T. T. "Real-time traffic flow forecasting using spectral analysis." IEEE Transactions on Intelligent Transportation Systems 13.2(2012):519-526.

[16] Yao, R., W. Zhang, and L. Zhang. "Hybrid methods for short-term traffic flow prediction based on ARIMA-GARCH model and wavelet neural network." Journal of Transportation Engineering Part A Systems 146.8(2020):04020086.

[17] Yu, Y., et al. "A low rank dynamic mode decomposition model for short-term traffic flow prediction." IEEE Transactions on Intelligent Transportation Systems 22.10(2021): 6547-6560.

[18] Wu, C. H., et al. "Travel time prediction with support vector regression." IEEE Transactions on Intelligent Transportation Systems 5.4(2004):276-281.

[19] Castro-Neto, Manoel, et al. "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions." Expert Systems with Applications 36.3(2009):6164-6173.

[20] Zhang, W., and Y. Liu. “Traffic forecasting using least squares support vector machines.” Transportmetrica 5.3(2009):193-213.

[21] Zhang, Y., and Y. Xie. "Forecasting of Short-Term Freeway Volume with v -Support Vector Machines." Transportation Research Record: Journal of the Transportation Research Board 2024.1(2007):92-99.

[22] Asif, M. T., et al. "Spatiotemporal Patterns in Large-Scale Traffic Speed Prediction." IEEE Transactions on Intelligent Transportation Systems 15.2(2013):1-11.

[23] Affiliation, K. K., M. P. Affiliation, and V. K. K. Affiliation. “Short term traffic flow prediction in heterogeneous condition using artificial neural network.” Transport 30.4(2015):397-405.

[24] Rilett, L., and D Park. "Direct forecasting of freeway corridor travel times using spectral basis neural networks." Transportation Research Record Journal of the Transportation Research Board 1752.1(2001):140-147.

[25] Jiang, X., and H. Adeli. “Dynamic wavelet neural network model for traffic flow forecasting.” Journal of Transportation Engineering 131.10(2005).

[26] Vlahogianni, E. I., M. G. Karlaftis, and J. C. Golias. "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach." Transportation Research Part C: Emerging Technologies 13.3(2005):211-234.

[27] Ma, X., et al. "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data." Transportation Research Part C: Emerging Technologies 54(2015):187-197.

[28] Tian, Y., et al. "LSTM-based traffic flow prediction with missing data." Neurocomputing 318(2018):297-305.

[29] Jeong, M. H., et al. "Highway Speed Prediction Using Gated Recurrent Unit Neural Networks." Applied Sciences 11.7(2021):3059.

[30] Fu, R., Z. Zhang, and L. Li. “Using LSTM and GRU neural network methods for traffic flow prediction.” In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, 2016.

[31] Han, M., and M. Xu. “Laplacian echo state network for multivariate time series prediction.” IEEE Transactions on Neural Networks and Learning Systems 29.1(2017):238-244.

[32] Lin, X., Z. Yang, and Y. Song. “Short-term stock price prediction based on echo state networks.” Expert Systems with Applications 36.3(2009):7313-7317.

[33] Shi, Z., and M. Han. “Support vector echo-state machine for chaotic time-series prediction.” IEEE Transactions on Neural Networks 18.2(2007):359-372.

[34] Wang, H., et al. “Echo state network based ensemble approach for wind power forecasting.” Energy Conversion and Management 201(2019):112188.

[35] An, Y., Q. Song, and X. Zhao. "Short-term traffic flow forecasting via echo state neural networks." In 2011 Seventh International Conference on Natural Computation, 2011.

[36] Ser, J. D., et al. "Road Traffic Forecasting using Stacking Ensembles of Echo State Networks." In 2019 IEEE Intelligent Transportation Systems Conference, 2019.

[37] Gallicchio, C., A. Micheli, and L. Pedrelli. "Design of deep echo state networks." Neural Networks 108 (2018):33-47.

[38] Ser, J. D., et al. "Deep echo state networks for short-term traffic forecasting: performance comparison and statistical assessment." In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems, 2020.

[39] Pholsena, K., L. Pan, and Z. Zheng. "Multi-node Mode Decomposition Based Deep Learning Model for Road Section Traffic Prediction." In 2019 IEEE Fourth International Conference on Data Science in Cyberspace, 2019.

[40] Tang, J., et al. "Traffic flow prediction based on combination of support vector machine and data denoising schemes." Physica A: Statistical Mechanics and its Applications 534(2019):120642.

[41] Tian, X., et al. "Hybrid short-term traffic flow prediction model of intersections based on improved complete ensemble empirical mode decomposition with adaptive noise." Advances in Mechanical Engineering 11.4(2019).

[42] Kim, E., et al. "Evaluating spatiotemporal features of kinematic waves using ensemble empirical mode decomposition." In Transportation Research Board 98th Annual Meeting, 2019.

[43] Wang, H., et al. "Empirical Mode Decomposition-Autoregressive Integrated Moving Average: Hybrid Short-Term Traffic Speed Prediction Model." Transportation Research Record Journal of the Transportation Research Board 2460.1(2014):66-76.

[44] Jiang, X., L. Zhang, and X. Chen. "Short-term forecasting of high-speed rail demand: a hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China." Transportation Research Part C: Emerging Technologies 44(2014):110-127.

[45] Zhang, N., et al. "Wavelet-HST: A Wavelet-based Higher-order Spatio-Temporal Framework for Urban Traffic Speed Prediction." IEEE Access 7(2019):118446-118458.

[46] Liu, R., et al. "Short-Term Passenger Flow Prediction Based on Wavelet Transform and Kernel Extreme Learning Machine." IEEE Access 7(2019): 158025-158034.

[47] Zhang, N., et al. "A Hybrid Traffic Speed Forecasting Approach Integrating Wavelet Transform and Motif-based Graph Convolutional Recurrent Neural Network." ArXiv abs/1904.06656 (2019).

[48] Sun, P., N. Aljeri, and A. Boukerche. "A Fast Vehicular Traffic Flow Prediction Scheme Based on Fourier and Wavelet Analysis." In 2018 IEEE Global Communications Conference, 2019.

[49] Sun, Y., B. Leng, and W. Guan. "A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system." Neurocomputing 166(2015):109-121.

[50] Tang, J., et al. "Traffic flow prediction based on combination of support vector machine and data denoising schemes." Physica A: Statistical Mechanics and its Applications 534(2019): 120642.

[51] Kashi, S. O.M., and M. Akbarzadeh. "A framework for short-term traffic flow forecasting using the combination of wavelet transformation and artificial neural networks." Journal of Intelligent Transportation Systems 23.1(2019):60-71.

[52] Wei, Y., and M. C. Chen. "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks." Transportation Research Part C Emerging Technologies 21.1(2012):148-162.

[53] Li, L., et al. "Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network." Applied Soft Computing 73(2018):921-932.

[54] Diao, et al. "A Hybrid Model for Short-Term Traffic Volume Prediction in Massive Transportation Systems." IEEE Transactions on Intelligent Transportation Systems 20.3(2019):935-946.

[55] Kim, T., and B. R. King. "Time series prediction using deep echo state networks." Neural Computing and Applications 32.23(2020):17769-17787.

[56] Huang, N. E., 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 454(1998): 903-995.

[57] Liu, H., et al. "A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks." Renewable Energy 48(2012):545-556.

[58] Wei, Y., and M. C. Chen. “Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks.” Transportation Research Part C: Emerging Technologies 21.1(2012): 148-162.
[59] Tong, W., et al. “Comparing the applications of EMD and EEMD on time-frequency analysis of seismic signal.” Journal of Applied Geophysics 83(2012):29-34.
[60] Wu, Z., and Huang, N. E. “Ensemble empirical mode decomposition: a noise-assisted data analysis method.” Advances in Adaptive Data Analysis 1.1(2009): 1-41.
[61] Torres, M. E., et al. “A complete ensemble empirical mode decomposition with adaptive noise.” In 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2011.
[62] Aghajani, A., R. Kazemzadeh, and A. Ebrahimi. “A novel hybrid approach for predicting wind farm power production based on wavelet transform, hybrid neural networks and imperialist competitive algorithm.” Energy Convers Manage 121(2016):2322-40.
[63] Zhao, X., and B. Ye. “Convolution wavelet packet transform and its applications to signal processing.” Digital Signal Process 20.5(2010):1352-1364.
[64] Gallicchio, C., A. Micheli, and L. Pedrelli. “Deep reservoir computing: a critical experimental analysis.” Neurocomputing 268(2017):87-99.
[65] Lukosevicius, M., and H. Jaeger. “Reservoir computing approaches to recurrent neural network training.” Computer Science Review 3.3(2009):127-149.
[66] Gallicchio, C., A. Micheli, and L. Pedrelli. “Design of deep echo state networks.” Neural Networks 108(2018):33-47.
[67] Song, Z., K. Wu, and J. Shao. “Destination prediction using deep echo state network.” Neurocomputing 406(2020):343-353.
[68] Yu, B., Y. Lee, and K. Sohn. “Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN).” Transportation Research Part C: Emerging Technologies 114(2020):189-204.
[69] Lv, M., et al. “Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction.” IEEE Transactions on Intelligent Transportation Systems 22.6(2021):3337-3348.
[70] Cui, Z., K. Henrickson, and R. Ke. “Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting.” IEEE Transactions on Intelligent Transportation Systems 21.11(2020):4883-4894.
[71] Zhao, L., et al. “Temporal Graph Convolutional Network for Urban Traffic Flow Prediction Method.” (2018).
[72] Kipf, T. N., and M. Welling. “Semi-supervised classification with graph convolutional networks.” (2016).
[73] Bruna, J. et al. “Spectral Networks and Locally Connected Networks on Graphs.” (2013).