

Short-term traffic flow prediction based on cuckoo search-wavelet neural network

Haibo Lv

N/A

Yuheng Kang

N/A

Zhou Shen (✉ shenzhou.research@gmail.com)

N/A

Research Article

Keywords: Short-term traffic flow prediction

Posted Date: August 3rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-778000/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Short-term traffic flow prediction based on cuckoo search-wavelet neural network

Haibo Lv, Yuheng Kang, and Zhou Shen*

Abstract

The nonlinear fluctuation and uncertainty that characterize urban traffic flow are well-known. An Improved Cuckoo Search-Wavelet Neural Network (ICS-WNN) prediction model for urban traffic flow is suggested in order to increase the accuracy of traffic flow predictions. After the original traffic flow data have been cleaned up and normalized, the traffic flow prediction network model is built by optimizing the wavelet neural network weights and wavelet shrinkage and translation factors based on the adaptive step size and discovery probability of the cuckoo algorithm, and then adding the neural network momentum factor. The traffic flow prediction network model is built in two stages. The results of the experimental simulations demonstrate that the ICS-WNN prediction algorithm has a better fit and accuracy than numerous common optimization prediction techniques, which is encouraging.

1 Introduction

China's urbanization is progressing at a rapid pace, and at the same time, traffic congestion in cities is becoming more and more severe [1, 2]. As the country's economy continues to grow at its current pace, the progress of urbanization will become increasingly visible [3–7]. In addition, because the urban traffic system is open, real-time, and complex, with uncertainty and non-linear characteristics, it cannot be solved by relying solely on road widening, new roads, and traffic restrictions to alleviate traffic congestion caused by the contradiction between traffic demand and traffic supply [8–11]. An efficient and dependable traffic flow forecast system can provide an effective basis for the best scheduling and travel of traffic vehicles, as well as help to reduce congestion on the roads [12–15].

Many different approaches of predicting traffic flow have been developed by both domestic and international researchers [16–23]. For example, the combined model of the ARMA model and the Kalman filter model has higher prediction accuracy when compared to the ARMA model and the Kalman filter model alone; combining deep belief network model (DBN) and support vector machine (SVM) for traffic flow prediction, using DBN to learn traffic flow and extract

*The authors declare no competing interests.

data feature values, and using support vector machine model for traffic flow prediction; combining deep belief network model (DBN) and support vector machine (SVM) for traffic flow prediction, using DBN to The state vector and distance metric are combined to form the K-nearest neighbor prediction model, which reflects traffic flow more accurately than the k-neighbor model that only considers the time dimension; the wavelet neural network prediction algorithm based on the particle swarm algorithm, which uses the variable acceleration factor to make the particle swarm algorithm converge to the glo algorithm; and the wavelet neural network prediction algorithm based on the particle swarm algorithm, which uses the variable acceleration factor to make the particle [24–30]. In this study, we used the genetic algorithm to optimize the parameters of the wavelet neural network prediction model, taking advantage of the global optimization and adaptive random search characteristics of the genetic algorithm to optimize the parameters of the WNN; cereal bird search algorithm; optimized BP neural network prediction algorithm; traffic flow sequence for phase space reconstruction after network learning; and finally, the CS algorithm. The outcomes of the predictions are more accurate since they reflect the changing trend of traffic flow.

This research proposes, based on the CS-WNN short-time traffic flow prediction model, an enhanced cuckoo search method for optimizing the wavelet neural network prediction model, which can be used to predict traffic flow in real time. During the later stages of the development process, the cuckoo search algorithm will have the disadvantages of a poor search speed and a low prediction accuracy. This algorithm’s step size and discovery probability P_a are dynamically modified according to the number of iterations and adaptations performed in order to optimize the basic cuckoo algorithm. Improved convolutional neural networks are used to initialize the parameters of wavelet neural networks, and the neural network momentum term is included to prevent convergence jitter and speed up network convergence, which is then used to construct the prediction model for traffic flow forecasting. When compared to many more prominent traffic flow prediction models, the simulation results show that the prediction model described in this study has a better fit and higher accuracy.

2 Traffic flow data preprocessing

2.1 Wavelet denoising

As mentioned above, noise infiltrates the signal acquisition process to varying degrees, and the presence of noise frequently obscures the information that should be conveyed by the signals themselves. The purpose of signal denoising is to highlight and enhance the signal itself while suppressing the effect of noise on it. To reduce data noise in the traffic flow samples that were used in this paper, wavelet transforms were applied to reduce the data noise. Generally speaking, wavelet denoising is divided into three steps: (1) wavelet decomposition of the original signal to obtain the high-frequency part and low-frequency part of the

signal; (2) threshold processing of the noise in the high-frequency part of the signal; and (3) wavelet reconstruction with the processed wavelet coefficients to obtain the denoised signal. It is proposed in this paper to apply heuristic wavelet thresholding denoising, and the wavelet transform with two decomposition layers is used to denoise the traffic flow data by employing the sym3 wavelet in the wavelet toolbox to perform the denoising.

2.2 Normalization

The traffic data collection device captures a huge number of traffic data samples of varying sizes, and in order to more simply and correctly converge to the ideal solution, the data set must be normalized before it can be used [31–33]. In this study, we normalize the original data by employing the `mapminmax` function, which is defined as The data is standardized in order to speed up the model convergence process and increase the accuracy of the final model. By default, `mapminmax` is normalized to the interval [-1,1], and the computation equation is written as follows:

$$y = \frac{(y_{\max} - y_{\min}) \times (x - x_{\min})}{x_{\max} - x_{\min}} + y_{\min} \quad (1)$$

where y is the normalized processed data, $y_{\max} = 1$, $y_{\min} = -1$; x is the current original data to be normalized; x_{\max} and x_{\min} are the maximum and minimum values of the original data to be processed in the data sample, respectively.

After the processed data are trained by the network and the prediction model, the prediction results are normalized using the `mapminmax('reverse', 'y', 'ps')` function, and the normalized data y is normalized according to the given data normalization processing mapping ps , to obtain the normal real traffic flow prediction data.

3 Traffic flow prediction model

3.1 Introduction to Wavelet Neural Network

Wavelet neural networks are organic combinations that follow the structure of a neural network and replace the activation function in the network with a wavelet function, i.e., a neural network based on the BPN topology, with the wavelet basis function as the transfer function of nodes in the hidden layer, and the signal propagates forward, while the error propagates backwards. How to combine the advantages of wavelet transform and neural network has always been a hot topic of discussion among scholars. Wavelet transform has time-frequency local characteristics, while neural networks have self-learning, self-adaptive, robustness, fault tolerance, and generalization ability. When compared to traditional neural networks, the wavelet neural network overcomes structural design blindness and has a greater learning ability, faster convergence speed, and higher prediction accuracy than the traditional neural network.

It is possible to partition the WNN structure into three layers: an input layer, a hidden layer, and a final output layer. Consider the following scenario: there are M nodes in the input layer, n nodes in the hidden layer, and N nodes in the output layer. So far, there is no scientific or universal formula for determining the number of nodes in the hidden layer; instead, the number of nodes in the hidden layer can only be determined by using the trial-and-error method, which is as follows:

$$n = \sqrt{M + N} + a \quad (2)$$

The number of nodes in the hidden layer ranges between 3 and 12, and the specific value must be compared to network training in order to determine which has the best prediction impact.

As shown in Figure 2, the connection weights between the input layer and the hidden layer are denoted by W_{ij} , the connection weights between the hidden layer and the output layer are denoted by W_{jk} , and the wavelet function is denoted by $\Phi(x)$. The excitation function of the neuron used for this paper is the Morlet wavelet, and its mathematical expression is given by

$$\varphi(x) = \cos(1.75x) \exp(-x^2/2) \quad (3)$$

Wavelet neural network hidden layer output formula

$$P(j) = \varphi \left(\left(\sum_{i=1}^M W_{ij} x_i - b_j \right) / a_j \right) \quad j = 1, 2, \dots, n \quad (4)$$

where $P(j)$ is the j th output of the hidden layer; a_j is the scaling factor of the wavelet basis function; and b_j is the translation factor of the wavelet basis function in equation.

The following is the output $Y_p(k)$ of the output layer of the wavelet neural network as calculated by

$$Y_p(k) = \sum_{j=1}^n W_{jk} P(j) \quad k = 1, 2, \dots, N \quad (5)$$

By employing the gradient descent method, it is possible to get the best possible correction for the wavelet basis function coefficients and network connection weights in the wavelet neural network. The correction process is as follows:

1. The error between the wavelet neural network output prediction $Y_p(k)$ and the actual result $Y_r(k)$ is

$$e_k = Y_p(k) - Y_r(k) \quad (6)$$

2. The weights W_{ij} of the connection between the input layer-implicit layer, the weights W_{jk} of the connection between the implicit layer-output layer,

the scaling factor a_j of the wavelet function and the translation factor b_j of the wavelet function are adjusted according to the prediction error e_k :

$$\begin{aligned}
W_{ij}^{(d+1)} &= W_{ij}^d + \Delta W_{ij}^{(d+1)} \\
W_{jk}^{(d+1)} &= W_{jk}^d + \Delta W_{jk}^{(d+1)} \\
a_j^{(d+1)} &= a_j^d + \Delta a_j^{(d+1)} \\
b_j^{(d+1)} &= b_j^d + \Delta b_j^{(d+1)}
\end{aligned} \tag{7}$$

In the equations above, the term with the superscript d represents the connection weights or coefficients of the wavelet basis function for the d th training; the term with the superscript $(d + 1)$ represents the connection weights or coefficients of the wavelet basis function for the $d+1$ th training; $\Delta W_{ij}^{(d+1)}$ represents the connection weights correction term between the $d + 1$ st training input layer and the hidden layer; $\Delta W_{jk}^{(d+1)}$ represents the $d + 1$ st connection weight correction term between the training hidden layer and the output layer; $\Delta a_j^{(d+1)}$ denotes the scaling factor correction term of the $d+1$ th training wavelet function; $\Delta b_j^{(d+1)}$ denotes the translation factor correction term of the $d+1$ th training wavelet function. Each correction term is calculated based on the network error e_k as follows, where η is the learning rate:

$$\begin{aligned}
\Delta W_{ij}^{(d+1)} &= -\eta \cdot e / \partial W_{ij}^{(d)} \\
\Delta W_{jk}^{(d+1)} &= -\eta \cdot \partial e / \partial W_{jk}^{(d)} \\
\Delta a_j^{(d+1)} &= -\eta \cdot \partial e / \partial a_j^{(d)} \\
\Delta b_j^{(d+1)} &= -\eta \cdot \partial e / \partial b_j^{(d)}
\end{aligned} \tag{8}$$

Considering the influence of the $d - 1$ st weight on the d th weight, the momentum term is added to the neural network to effectively suppress the network from trapping in local minima. After adding the momentum term, the formulae for calculating each connection weight and wavelet function coefficients in the first iteration training network still use the above equations, and when the number of training is greater than or equal to 2, each network connection weight and wavelet basis function coefficients are calculated using

$$\begin{aligned}
W_{ij}^{(d+1)} &= W_{ij}^d + \Delta W_{ij}^{(d+1)} + \alpha \left(W_{ij}^d - W_{ij}^{(d-1)} \right), \\
W_{jk}^{(d+1)} &= W_{jk}^d + \Delta W_{jk}^{(d+1)} + \alpha \left(W_{jk}^d - W_{jk}^{(d-1)} \right) \\
a_j^{(d+1)} &= a_j^d + \Delta a_j^{(d+1)} + \alpha \left(a_j^d - a_j^{(d-1)} \right), \\
b_j^{(d+1)} &= b_j^d + \Delta b_j^{(d+1)} + \alpha \left(b_j^d - b_j^{(d-1)} \right)
\end{aligned} \tag{9}$$

3.2 Introduction to cuckoo search and improvement

3.2.1 Cuckoo search

The cuckoo search algorithm is an algorithm that efficiently solves optimization problems by simulating the parasitic brood of certain species of cuckoos. The cuckoo search algorithm requires setting three ideal states:

1. Only one cuckoo egg is laid per cuckoo at a time and the eggs are placed randomly in a nest;
2. The nest with the best cuckoo egg will be kept to the next generation;
3. The number of nests is fixed, and the probability that a cuckoo egg is found in a nest is P_a , taking values in the range (0,1).

It can be understood that nest=egg=solution, the process of cuckoo searching for nests to lay eggs is the process of searching for solutions, and the quality of the nest represents the good or bad solution.

At the same time, the cuckoo search algorithm location update using the relevant Levy flight random path mechanism, Levy flight to generate random numbers should contain two steps: the choice of random direction and obey the Levy distribution step generation, the random distribution of direction should obey the uniform distribution, the most effective and direct way to generate the step is to use the Mantegna algorithm to achieve a symmetric levy stable distribution. The formula for the Levy flight to update the bird's nest location is defined as:

$$X_{t+1} = X_t + \alpha \otimes \text{Levy}(\beta) \quad (10)$$

where X_t denotes the nesting position at generation t ; X_{t+1} denotes the nesting position at generation $t + 1$, α denotes the step scaling factor; $\text{Levy}(\beta)$ denotes the Levy random path, and the relationship between the Levy random path and time t obeys the Levy distribution

$$\text{levy}(\beta) \sim u = t^{-\lambda}, 1 < \lambda \leq 3 \quad (11)$$

In addition, a portion of the host birds found cuckoo eggs with probability P_a , and the location path of abandoning the nest and building a new nest was chosen to prefer a random swim with the calculation formula

$$X_i^{t+1} = X_i^t + r \otimes (X_j^t - X_k^t) \quad (12)$$

where X_i^{t+1} is the i -th nesting position in generation $t + 1$, X_i^t is the i -th nesting position in generation t , X_j^t and X_k^t are two random nesting positions in generation t , and r is a random number obeying uniform distribution.

3.2.2 Improved cuckoo search

The basic cuckoo search algorithm has a strong global search capability in the early stage, but weakens the local search capability, and tends to converge to local extremes in the later stage, leading to slower search and poor prediction accuracy. In order to improve the prediction accuracy, this paper dynamically adjusts the step size and discovery probability, which vary with the number of iterations of the algorithm. The following equations are used to update the discovery probability and step length:

$$\begin{aligned}
 P_{ai}(t) &= nP_{a \max} \times \exp(c) \times \frac{f_i}{\sum_{j=1}^n f_j}, \\
 c &= \frac{t}{T} \times \ln \left(\frac{P_{a \min}}{P_{a \max}} \right), \\
 \alpha_i(t) &= n \left[\alpha_{\max} - \frac{t}{T} (\alpha_{\max} - \alpha_{\min}) \right] \times \frac{f_i}{\sum_{j=1}^n f_j}.
 \end{aligned} \tag{13}$$

where $P_{ai}(t)$ denotes the probability that the i th nest in the t th generation is discovered, the maximum discovery probability $P_{a \max}$, the minimum discovery probability $P_{a \min}$, such that $P_{a \max} = 1$, $P_{a \min} = 0.005$; T denotes the maximum number of iterations, t is the current number of iterations, n is the number of nests; $\alpha_i(t)$ is the update step of the i th nest position in the t th generation, α_{\min} and α_{\max} respectively are the maximum and minimum steps, such that $\alpha_{\min} = 0.05$ and α_{\max} ; f_i is the adaptation value of the i th nest, and f_j is the adaptation value of any one nest.

The basic steps of the ICS-WNN algorithm are as follows:

1. Import the original traffic flow data for pre-processing, and split the processed data into training data and test data;
2. Initialize the parameters of the wavelet neural network, train the neural network for several times, finally determine the values of the parameters and set the maximum number of iterations of the neural network to 500;
3. Initialize the number of nests N_{nest} , P_a is dynamically adjusted by exponentially decreasing, the maximum number of iterations T_{max} is set to 150, and each nest location W_i has l ($l = M \times n + N \times n + n + n + n$, M is the number of nodes in the input layer, n is the number of nodes in the hidden layer, N is the number of nodes in the output layer) parameters. A linear decreasing step strategy is used to update the location and path, and the training is performed according to the initial values of each parameter to calculate the prediction value and prediction error for each nest, and find the nest location with the smallest prediction error among N_{nest} nests, mark it as the best nest W_{best} and keep it to the next generation;
4. In the next training, the location and path are updated for the nests in 3) other than the best nest W_{best} . The error e_{k+1} between the predicted

and actual values of bird nest locations is calculated by wavelet neural network and compared with the prediction error e_k of other nests in the last training result, if e_{k+1} is greater than e_k , the k th nest location is retained; if e_{k+1} is less than e_k , the k th+1st training nest location is retained, thus finding the contemporary best nest location W_{new} ;

5. If the random number $r > P_a$, this group of bird nest locations is discarded, and the location and path of the new nest are constructed, and the update is done by using the preferred random tour. The prediction error is calculated by wavelet neural network and compared with each prediction error in the better bird nest location W_{new} in step 4), and the bird nest location with smaller prediction error then replaces the bird nest location with larger prediction error, thus obtaining the contemporary better bird nest location W'_{new} ;
6. The optimal bird nest location in W'_{new} is marked as W_{best} , and if the number of iterations or the prediction accuracy is not reached, it returns to step 4) to continue the optimization search, and vice versa to output the current W_{best} ;
7. The finalized W_{best} is split into each connection weight, the scaling factor and translation factor of wavelet basis function, respectively. The processed data are imported into the wavelet neural network and trained in the network, and when the conditions are met, the network training is stopped and the values of the parameters at this time are saved. The conditions for the end of training are: reaching the maximum number of iterations or reaching the prediction accuracy;
8. Input test data for actual traffic flow prediction. Compare the predicted traffic flow results with the actual traffic flow and calculate the value of each evaluation index.

4 Case studies

The experimental hardware platform is a quad-core 7th generation Core i7 processor with 8GB DDR4 memory, and the software environment is MATLAB R2014b. The experimental data is selected from the traffic flow data of a lane in Stockton, San Joaquin County, California, USA, recorded from April 4, 2011 to April 15, 2011 for a total of 10 weekdays with a 15-minute interval. The traffic flow data were recorded for 10 weekdays at 15-minute intervals from April 4, 2011 to April 15, 2011.

In the prediction model, the learning rate of each connection weight is 0.01, the learning rate of the translation and scaling factors of wavelet function is 0.0015, the momentum factor is 0.4, the number of nests is 20, the P_a is dynamically adjusted by exponentially decreasing, and the step size is linearly decreasing.

In order to analyze the prediction model proposed in this paper, four evaluation metrics are selected to analyze the prediction results, namely mean absolute percentage error (MAPE), mean absolute error (MAE), mean coefficient of equality (EC) and root mean square error (RMSE).

In order to verify the prediction effect of the algorithm proposed in this paper, PSO-WNN algorithm, GA-WNN algorithm, CS-BPNN algorithm, CS-WNN algorithm and the algorithm proposed in this paper were selected to compare the traffic flow prediction results, and the results are shown in Table 1.

Model	MAPE	MAE	RMSE	EC
ICS-WNN	5.773	10.521	12.846	0.973
CS-WNN	6.598	11.201	14.456	0.970
PSO-WNN	7.306	12.183	15.305	0.969
GA-WNN	7.850	13.055	16.131	0.967
CS-BPNN	8.898	13.889	17.404	0.965

The mean coefficients (goodness of fit) of several algorithms in the table are above 0.95, and the higher the goodness of fit indicates that the prediction effect is better, so the prediction results using these algorithms are more meaningful. The goodness-of-fit of the proposed algorithm is 0.97386, which is the highest goodness-of-fit value among several prediction algorithms, indicating that the prediction effect of the proposed algorithm is better than several other mainstream prediction algorithms.

5 Conclusion

The improved cuckoo search algorithm is used to initialize the weights of each connection of wavelet neural network with the scaling factor and translation factor of wavelet function in order to overcome the shortcomings of the cuckoo search algorithm such as low accuracy and slow search speed in the later stage; then the improved cuckoo search algorithm is used to initialize the weights of each connection of wavelet neural network with the scaling factor and translation factor of wavelet function in order to overcome the shortcomings of the cuckoo search algorithm such While building the ICS-WNN prediction model, it is necessary to include a momentum element in order to avoid oscillation during the network's training. By comparing the evaluation index values of many prediction algorithms, it has been demonstrated that the prediction model provided in this study has higher prediction accuracy and better fitting ability than the other prediction algorithms. Traffic flow is particularly susceptible to the influence of the external environment, and the next work will explore the investigation of prediction algorithms with additional input factors in order to further increase the accuracy of traffic flow prediction.

References

- [1] K. W. Chan, “Fundamentals of china’s urbanization and policy,” *China Review*, pp. 63–93, 2010.
- [2] Y. Li, L. Jia, W. Wu, J. Yan, and Y. Liu, “Urbanization for rural sustainability—rethinking china’s urbanization strategy,” *Journal of Cleaner Production*, vol. 178, pp. 580–586, 2018.
- [3] J. J. Q. Yu, C. Markos, and S. Zhang, “Long-term urban traffic speed prediction with deep learning on graphs,” *IEEE Transactions on Intelligent Transportation Systems*, 2021, in press.
- [4] M. B. Arvin, R. P. Pradhan, and N. R. Norman, “Transportation intensity, urbanization, economic growth, and co2 emissions in the g-20 countries,” *Utilities Policy*, vol. 35, pp. 50–66, 2015.
- [5] C. Zhang, Q. Zhang, S. Yu, J. J. Q. Yu, and X. Song, “Complicating the social networks for better storytelling: an empirical study of chinese historical text and novel,” *IEEE Transactions on Computational Social Systems*, 2021, in press.
- [6] O. Kotavaara, H. Antikainen, and J. Rusanen, “Urbanization and transportation in finland, 1880–1970,” *Journal of Interdisciplinary History*, vol. 42, no. 1, pp. 89–109, 2011.
- [7] J. J. Q. Yu, “Two-stage request scheduling for autonomous vehicle logistic system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1917–1929, May 2019.
- [8] R. Arnott, A. De Palma, and R. Lindsey, “Does providing information to drivers reduce traffic congestion?” *Transportation Research Part A: General*, vol. 25, no. 5, pp. 309–318, 1991.
- [9] K. K. Dewan and I. Ahmad, “Carpooling: a step to reduce congestion,” *Engineering Letters*, vol. 14, no. 1, pp. 61–66, 2007.
- [10] C. Pasquale, I. Papamichail, C. Roncoli, S. Sacone, S. Siri, and M. Papageorgiou, “Two-class freeway traffic regulation to reduce congestion and emissions via nonlinear optimal control,” *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 85–99, 2015.
- [11] J. J. Q. Yu, “Sybil attack identification for crowdsourced navigation: a self-supervised deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, 2020, in press.
- [12] C. P. Green, J. S. Heywood, and M. N. Paniagua, “Did the london congestion charge reduce pollution?” *Regional Science and Urban Economics*, vol. 84, p. 103573, 2020.
- [13] C. Zhang, S. Zhang, J. J. Q. Yu, and S. Yu, “FASTGNN: A topological information protected federated learning approach for traffic speed forecasting,” *IEEE Transactions on Industrial Informatics*, 2021, in press.
- [14] X. Zou, S. Zhang, C. Zhang, J. J. Q. Yu, and E. Chung, “Long-term origin-destination demand prediction with graph deep learning,” *IEEE Transactions on Big Data*, 2021, in press.
- [15] J. J. Q. Yu and A. Y. S. Lam, “Autonomous vehicle logistic system: Joint routing and charging strategy,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2175–2187, Jul. 2018.
- [16] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, “Predicting traffic flow using bayesian networks,” *Transportation Research Part B: Methodological*, vol. 42, no. 5, pp. 482–509, 2008.
- [17] N. G. Polson and V. O. Sokolov, “Deep learning for short-term traffic flow prediction,” *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.
- [18] J. J. Q. Yu, “Travel mode identification with GPS trajectories using wavelet transform and deep learning,” *IEEE Transactions on Intelligent Transportation Systems*, 2020, in press.
- [19] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: a deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.
- [20] B. Hamner, “Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow,” in *2010 IEEE International Conference on Data Mining Workshops*. IEEE, 2010, pp. 1357–1359.

- [21] J. J. Q. Yu, "Citywide traffic speed prediction: a geometric deep learning approach," *Knowledge-Based Systems*, p. 106592, Nov. 2020.
- [22] J. Dong and H. S. Mahmassani, "Stochastic modeling of traffic flow breakdown phenomenon: Application to predicting travel time reliability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1803–1809, 2012.
- [23] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: a federated learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751–7763, Aug. 2020.
- [24] J. J. Q. Yu, "Semi-supervised deep ensemble learning for travel mode identification," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 120–135, Mar. 2020.
- [25] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting," *Journal of transportation engineering*, vol. 131, no. 10, pp. 771–779, 2005.
- [26] C. Zhang, J. J. Q. Yu, and Y. Liu, "Spatial-temporal graph attention networks: a deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166 246–166 256, 2019.
- [27] H.-F. Yang, T. S. Dillon, and Y.-P. P. Chen, "Optimized structure of the traffic flow forecasting model with a deep learning approach," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2371–2381, 2016.
- [28] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3940–3951, Oct. 2019.
- [29] C. Zhang, S. Zhang, J. J. Q. Yu, and S. Yu, "An enhanced motif graph clustering-based deep learning approach for traffic forecasting," in *Proc. IEEE Global Communications Conference*. Taipei, Taiwan: IEEE, 2020, pp. 1–6.
- [30] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 1, pp. 124–132, 2006.
- [31] J. J. Q. Yu, W. Yu, and J. Gu, "Online vehicle routing with neural combinatorial optimization and deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3806–3817, Oct. 2019.
- [32] H. Chen and S. Grant-Muller, "Use of sequential learning for short-term traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 5, pp. 319–336, 2001.
- [33] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303–321, 2002.