Transportation Research Part C 115 (2020) 102620

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction☆

Zhiyong Cui[a], Ruimin Ke[a], Ziyuan Pu[a], Xiaolei Ma[b], Yinhai Wang[a,*]

[a] Deparment of Civil and Environmental Engineering, University of Washington, Seattle, WA, USA
[b] School of Transportation Science and Engineering, Beihang University, Beijing, China

ARTICLE INFO

Keywords:
Traffic forecasting
Deep learning
Graph wavelet
Recurrent neural network
Sparsity
Interpretability

ABSTRACT

Network-wide traffic forecasting is a critical component of modern intelligent transportation systems for urban traffic management and control. With the rise of artificial intelligence, many recent studies attempted to use deep neural networks to extract comprehensive features from traffic networks to enhance prediction performance, given the volume and variety of traffic data has been greatly increased. Considering that traffic status on a road segment is highly influenced by the upstream/downstream segments and nearby bottlenecks in the traffic network, extracting well-localized features from these neighboring segments is essential for a traffic prediction model. Although the convolution neural network or graph convolution neural network has been adopted to learn localized features from the complex geometric or topological structure of traffic networks, the lack of flexibility in the local-feature extraction process is still a big issue. Classical wavelet transform can detect sudden changes and peaks in temporal signals. Analogously, when extending to the graph/spectral domain, graph wavelet can concentrate more on key vertices in the graph and discriminatively extract localized features. In this study, to capture the complex spatial-temporal dependencies in network-wide traffic data, we learn the traffic network as a graph and propose a graph wavelet gated recurrent (GWGR) neural network. The graph wavelet is incorporated as a key component for extracting spatial features in the proposed model. A gated recurrent structure is employed to learn temporal dependencies in the sequence data. Comparing to baseline models, the proposed model can achieve state-of-the-art prediction performance and training efficiency on two real-world datasets. In addition, experiments show that the sparsity of graph wavelet weight matrices greatly increases the interpretability of GWGR.

1. Introduction

Traffic pattern recognition is critical for modern intelligent transportation systems (ITS) and the planning for smart cities. A comprehensive recognition of historical urban traffic patterns is not only capable of identifying the recurrent congestions and bottlenecks of urban traffic networks, but also able to largely enhance the forecasting of future traffic states. As a component of ITS, accurate network-wide traffic prediction is the prerequisite for dynamic route planning and traffic assignment optimization. The growing need for short-term prediction of traffic parameters embedded in ITS has led to a great deal of research on traffic forecasting in the last three decades (Vlahogianni et al., 2004). Before the rise of artificial intelligence, traffic forecasting methods have been

☆ This article belongs to the Virtual Special Issue on "Machine learning".

* Corresponding author.
E-mail address: yinhai@uw.edu (Y. Wang).

https://doi.org/10.1016/j.trc.2020.102620
Received 1 May 2019; Received in revised form 25 December 2019; Accepted 11 March 2020
Available online 10 April 2020

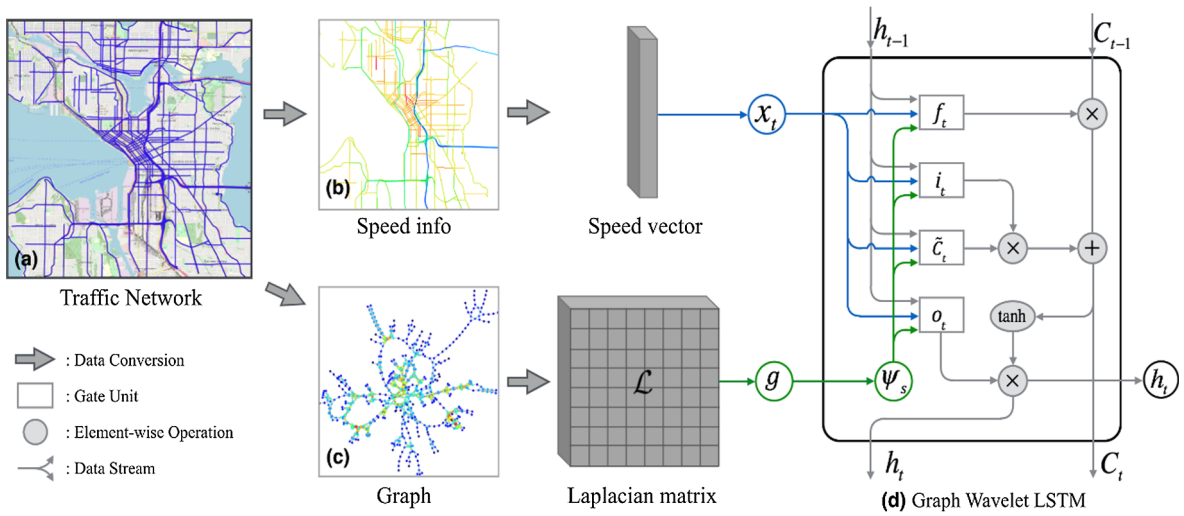


Fig. 1. Demonstration of model framework. (a) Urban traffic network in downtown Seattle. (b) Speed information of roadway segments illustrated by various colors. (c) Graph structure converted from the traffic network. (d) Structure of a graph wavelet LSTM unit at time t, in which g is the kernel function and Ψ_s is the graph wavelet matrix.

gradually shifting from traditional statistical models to computational intelligence, or say machine learning methods (Vlahogianni et al., 2014). With the exponentially increase in the volume of traffic data and the computational capability, a large amount of deep learning models, including recurrent neural network (RNN) (Ma et al., 2015), convolutional neural network (CNN) (Ma et al., 2017), stacked autoencoder (Lv et al., 2015), and their combinations (Li et al., 2017; Yu et al., 2018) were adopted for short-term traffic forecasting in recent years.

Much previous research focusing on traffic prediction only studied a roadway segment or several consecutive segments on a corridor. It is proved that using the information of multiple sensors/locations can help prediction models track short-term trends and enhance prediction performance (Li et al., 2015). Thus, in order to enhance traffic prediction performance and bring the power of artificial intelligence into real applications in the transportation field, it is inevitable to take large-scale traffic networks as the study areas. Due to the complex structure of traffic networks, there are several obvious hurdles in the traffic forecasting process.

The first question is how to represent the complex structure of a roadway network accurately? An example of a complex traffic network is shown in Fig. 1 (a). Previous studies attempted to convert the traffic states of sensing locations in a roadway network into a 2D spatial-temporal matrix (Ma et al., 2017) or convert the geometrical structure of urban roadway networks as colored images (Yu et al., 2017) for learning traffic states' features, where an example is shown in Fig. 1(b). However, in these ways, the topology of a roadway network cannot be adequately represented and the relationship of the adjacent roadway segments can hardly be comprehensively learned. To address this issue, many studies (Cui et al., 2019; Yu et al., 2018; Zhang et al., 2018; Li et al., 2017) have proposed a more elegant way to consider the traffic network as a graph consisting of vertices and edges denoting roadway segments and intersections, respectively, as shown in Fig. 1(c). In this way, the topology of a traffic network can be comprehensively represented by a graph.

Second, given considering the traffic network as a graph, designing an effective feature extraction process for the non-linear structured data is also challenging. Some powerful mathematical tools, such as graph convolution operator (Henaff et al., 2015; Bruna et al., 2013), have been adopted to integrate graph features into the traffic forecasting problems (Li et al., 2017; Cui et al., 2019). However, the original form of graph convolution is not well-localized, which means that, with respect to a centered node, the graph convolution cannot learn features exclusively from its neighboring nodes within a specific scale. Although Defferrard et al. (2016) proposed a fast localized spectral filter to enable the localization of the graph convolution, the receptive field is still not flexible. The neighboring field of a centered node is strictly confined by a ball of a designated radius covering a corresponding amount of hops of neighboring nodes.

Thus, the third issue is the lack of methods to enable the flexible local feature extraction process in a traffic prediction model. As shown in Li et al. (2015), traffic time series have long-term and short-term trends. Generally, the traffic state of a road is mainly affected by its neighboring roads. However, some key roadway segments, such as traffic bottlenecks or the ones with critical incidents, can highly affect the operational performance of the entire traffic network. Thus, much attention should be discriminatively paid to these segments in the feature extraction process. The classical wavelet transform inherently with the localization property can capture the sudden changes and detect peaks in a signal. Considering the spectral graph convolution is defined based on Fourier transform (Shuman et al., 2012), analogously, wavelet transform can be extended to the spectral domain as the graph wavelet (Hammond et al., 2011) to overcome the localization problem in the graph convolution. Graph wavelets are localized in the vertex domain, reflecting the information diffusion centered at each node. Based on the graph wavelet theory (Hammond et al., 2011), Xu et al. first proposed the graph wavelet neural network (Xu et al., 2019) to solve semi-supervised classification problems. In the traffic modeling process, the graph wavelet can also be used to flexibly extract comprehensive features and automatically concentrate more

on critical segments in the traffic network. Thus, we adopt the graph wavelet as a component of the proposed neural network in this study.

In addition, a traffic network related dataset not only contains geospatial information but also forms as temporal sequences. RNN and its variants, like long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014) neural networks, have been proved to be the superior methods to deal with sequence data in multiple fields. Since the traffic prediction is based on a long stream of historical data, the vanishing gradient problem often occurs in the vanilla RNN. To overcome this problem, a gated structure should be adopted in the prediction model.

To address these issues, we propose a new model for network-scale traffic learning and prediction. In this study, we learn the traffic network as a graph. To overcome the localization problem in graph convolution and make the receptive field more flexible, we adopt the graph wavelet as a key component of the proposed neural network. The graph wavelet is localized in the vertex domain. The graph wavelet operation considers both the traffic states of a road network and the underlying topology structure of the network. In order to learn the spatial-temporal features from the traffic network and fulfill traffic forecasting, we propose a graph wavelet gated recurrent neural network (GWGR). As tested on two real-world traffic datasets, the proposed GWGR model achieves better forecasting accuracy with fewer weight parameters comparing to existing benchmark models. The contributions of this study are summarized as follows:

1. We learn the traffic network as a graph and incorporate the graph wavelet as a key component to extract well-localized features from the traffic network based graph. Comparing to graph convolution, graph wavelet is pretty flexible with no need to specify the neighboring area in the topological graph structure for feature extraction. To the best of our knowledge, this is the first time that a graph wavelet based neural network is utilized for traffic forecasting.
2. We propose a graph wavelet gated recurrent neural network to learn from the spatial-temporal traffic network data, in which the graph wavelet operators act as filters in the gates of the recurrent neural network.
3. The proposed graph wavelet gated recurrent neural network can achieve superior prediction performance with fewer weight parameters and higher training efficiency, comparing to many existing models.
4. The quantifiable learned weights of the graph wavelets turn out to be sparse. This sparse property of learned weights can largely enhance the interpretability of the model and help to identify the key roadway links in the traffic network.

The rest of this paper is organized as follows. Section 2 discusses the existing literature. Section 3 defines multiple graph wavelet related concepts and describes the methodology in detail. The experimental results are shown in Section 4. Finally, we conclude the paper in Section 5.

2. Literature review

2.1. Classical models based traffic prediction

Previous traffic prediction methods can be categorized into two main groups, i.e. parametric approaches and nonparametric approaches (Lv et al., 2015). Parametric traffic prediction approaches are developed based on a predefined model structure with several certain theoretical assumptions, and parameters are calibrated using historical data (Smith et al., 2002). A variety of parametric traffic prediction approaches were proposed, including many variants of autoregressive integrated moving average (ARIMA) models (Williams, 2001), parametric Kalman filtering models (Okutani and Stephanedes, 1984), and other types of time-series models (Ghosh et al., 2009). In order to accommodate the stochastic and nonlinear nature of traffic flow, nonparametric approaches were also widely adopted for traffic prediction or traffic data imputation, including k-nearest neighbor (k-NN) methods (Chang et al., 2012), support vector regression (SVR) (Wu et al., 2004), Bayesian network approaches (Sun et al., 2006), and tensor decomposition approach (Chen et al., 2019). In spite of classical traffic prediction models are well studied and applied, it is pretty difficult for these models to deal with huge amount of large-scale network-wide traffic data.

2.2. Deep learning models based traffic prediction

Traffic prediction is widely studied in recent years with the rise of artificial intelligence. Existing literature shows that deep neural network models achieve superior prediction performance comparing to classical traffic prediction models. Due to traffic prediction is based on historical sequence data, most of the existing deep learning models (Ma et al., 2015; Cui et al., 2017; Wang et al., 2019; Pu et al., 2019) are built on RNN and its variants like LSTM and GRU. Because of the spatiotemporal characteristics of traffic data, a large amount of other types of deep learning models, including convolutional neural network (CNN) (Ma et al., 2017), stacked autoencoder (Lv et al., 2015), generative adversarial network (GAN) (Liang et al., 2018), and capsule network (Ma et al., 2018), and multiple combinations (Cui et al., 2017; Liao et al., 2018; Lv et al., 2018; Wu et al., 2018; Ke et al., 2019) of these models were utilized to capture spatial features and estimate traffic states. However, there is still much room for the aforementioned models to improve their feature extraction when taking the complicated nature of the traffic network structure into consideration.

2.3. Graph based deep learning models for traffic prediction

Due to many real-world data are non-linear structured data, graph based deep learning models attracted much attention in recent

years (Zhou et al., 2018). Roadway network based traffic data as a representative graph structured data can be effectively analyzed by taking many advantages of graph based models. Graph embedding methods (Yao et al., 2018) and graph attention networks (Zhang et al., 2018) have been adopted as components of neural network structures to deal with spatiotemporal traffic prediction problems. Based on the spectral graph theory (Shuman et al., 2012), spatiotemporal graph convolution neural network (Yu et al., 2018), diffusion convolutional recurrent neural network (Li et al., 2017), graph convolutional neural network with data-driven graph filter (Lin et al., 2018), etc. are proposed for traffic forecasting. A traffic graph convolutional recurrent neural network (Cui et al., 2019) incorporating the physical properties of roadways is also proposed for improving forecasting accuracy and enhancing the interpretability of the model. However, the interpretation and flexible localization of the feature extraction process in these graph convolution based models are not well addressed and discussed. A recent work (Xu et al., 2019) proposed the graph wavelet neural network to implement efficient convolution on graph data to solve semi-supervised classification problems. In this study, inspired by the graph wavelet neural network, we attempt to address those issues by adopting the graph wavelet as a component of the proposed gated recurrent model.

3. Methodology

3.1. Notions

The spatial-temporal traffic data collected by sensors are normally denoted as $X = [x_0, x_1, \cdots, x_t, \cdots, x_{T-1}] \in \mathbb{R}^{T\times N}$, in which T is the length of time steps and N is the number of traffic sensing locations. Each element x_t^n in X denotes the traffic status at t-th observation on the n-th location. Due to most of traffic sensing locations are connected by bidirectional roadways, the traffic network can be considered as an undirected graph consisting of vertices and edges representing sensing locations and connecting links, respectively. The graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ consisting of N vertices or nodes $v_j \in \mathcal{V}$ and their linking edges $(v_i, v_j) \in \mathcal{E}$. For clearance, graph nodes refer to vertices and they might be used interchangeably in this paper. The adjacency matrix $A \in \mathbb{R}^{N\times N}$ is used to describe the connectedness of vertices, in which element $A_{i,j} = A_{j,i} = 1$ if vertices i and j are connected, otherwise $A_{i,j} = 0$ ($A_{i,i} = 0$). Then a diagonal graph degree matrix $D \in \mathbb{R}^{N\times N}$ describing how many edges are attached to each vertex can be obtained by $D_{i,i} = \sum_{j=1}^{N} A_{i,j}$. The connectivity of the graph vertices can also be encoded in the graph Laplacian matrix $\mathcal{L}$, which is essential for spectral graph analysis. The combinatorial definition is $\mathcal{L} = D - A$ and the normalized Laplacian matrix is defined as $\mathcal{L} = I_N - D^{-1/2}AD^{-1/2}$, where $I_N \in \mathbb{R}^{N\times N}$ is the identity matrix. Due to $\mathcal{L}$ is a symmetric positive semidefinite matrix, it can be diagonlaized as $\mathcal{L} = U\Lambda U^T$ by eigenvector matrix $U = [u_0, u_1, \cdots, u_{N-1}] \in \mathbb{R}^{N\times N}$ and its corresponding diagonal eigenvalue matrix $\Lambda = \mathrm{diag}(\lambda_0, \lambda_1, \cdots, \lambda_{N-1}) \in \mathbb{R}^{N\times N}$ satisfying $\mathcal{L}u_i = \lambda_i u_i$.

3.2. Graph Fourier transform and graph convolution

The Fourier transform of a signal in the temporal domain can be regarded as the linear combination over a set of Fourier basis with different frequencies in the frequency domain. The larger the weight of a Fourier basis is, the more contribution the specific basis makes. Similarly, a graph signal in the vertex domain can also be transformed into a combination of a set of base in the frequency/spectral domain. The spectrum of graph Laplacian $\mathcal{L}$ carries a notion of frequency represented by the eigenvectors that the eigenvectors associated with higher eigenvalues generally have more zero crossings (Shuman et al., 2013). Taking the eigenvectors in U as the Fourier basis, the graph Fourier transform on graph signal vector $x \in \mathbb{R}^N$ can be defined as $\hat{x} = U^T x$ and its inverse as $x = U\hat{x}$, where $\hat{x}$ is the graph signal in the frequency domain. According to the convolution theorem (Bracewell and Bracewell, 1986), the Fourier transform of a convolution is the element-wise product of Fourier transform, and thus, the convolution operator $\otimes_{\mathcal{G}}$ on graph $\mathcal{G}$ can be generalized as:

$$f \otimes_{\mathcal{G}} x = U(\hat{f} \odot \hat{x}) = U((U^T f) \odot (U^T x))\# \tag{1}$$

where f is a filter function that can be considered as the convolutional kernel and $\odot$ is the element-wise multiplication operator. According to matrix multiplication rule, the $U^T f$ can be replaced by a diagonal matrix $\hat{f}(\Lambda_\theta)$ and Eq. (1) can be represented by

$$f \otimes_{\mathcal{G}} x = U\hat{f}(\Lambda_\theta)U^T x\# \tag{2}$$

where $\hat{f}$ is the filter function in the frequency domain and Λ_θ is a diagonal parameter matrix. When applying to deep learning models, the graph convolution of x is normally defined as $U\Lambda_\theta U^T x$, and here, Λ_θ is the diagonal weight matrix that should be learnt during the training process.

The spectral graph convolution operator has been adopted in many neural network structures (Henaff et al., 2015; Bruna et al., 2013). In those graph convolution layers, the diagonal matrix Λ_θ acts as a learnable filter function and its diagonal elements are all learnable weight parameters. However, due to the graph convolution operation is defined based on Fourier basis U, the convolution result on one vertex of a graph signal comes from all vertices, i.e. the receptive field of the graph convolution operator covers the whole graph structure. Thus, the graph convolution is not well-localized in the vertex domain. To overcome this limitation, a polynomial filter is proposed (Defferrard et al., 2016) to conduct graph convolution on one vertex from the signals of its exactly k hops of neighboring vertices, functioning as a localized receptive field in conventional CNNs. The filter is defined as

$$g_\theta = \sum_{k=0}^{K-1} \theta_k \Lambda^k\# \tag{3}$$

where $\theta \in \mathbb{R}^K$ is the learnable polynomial coefficients and Λ^k is the k-order Laplacian diagonal eigenvalue matrix. In this case, the graph convolution on the Laplacian is K-localized. To apply the graph convolution in the transportation network learning and take the roadway physical properties into consideration, a free flow reachability matrix is incorporated in the traffic graph convolution (Cui et al., 2019).

However, in those existing methods, the hyperparameter K is still required to be specified to decide the size of graph convolution receptive field, i.e. the hops of neighborhood nodes that influence a centering node. Further, the receptive field confined by K is not flexible enough considering that key vertices in a graph are normally not evenly distributed.

3.3. Wavelet transform

Fourier transform can decompose a signal in the temporal domain into a combination of frequencies, while wavelet transform can indicate when and where the frequencies contribute more, if a signal varies with time. The wavelet transform adopts a wavelet prototype function, i.e. mother wavelet, to cut up signal data into different frequency components and studies each component with a resolution matched to its scale (Bruna et al., 2013). A classical wavelet $\psi_{s,a}$ of a function at scale s and location a is constructed from the translated and scaled mother wavelet ψ:

$$\psi_{s,a}(x) = \frac{1}{s}\psi\left(\frac{x-a}{s}\right)\# \tag{4}$$

And the wavelet coefficients $W_f(s, a)$ is acquired by convolving the input $f(x)$ with wavelet:

$$W_f(s, a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)dx \tag{5}$$

in which $(\bullet)^*$ denotes complex conjugate and ψ^* equals to ψ for real-valued and even the mother wavelet. Then the classical wavelet transform operator T^s for a scale s at location a can be denoted by $T^s f(a) = W_f(s, a)$. Letting $\bar{\psi}_s(x) = \frac{1}{s}\psi^*\left(\frac{-x}{s}\right)$, the operator T^s can be written as:

$$\begin{aligned} T^s f(a) &= \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)dx \\ &= \int_{-\infty}^{\infty} \frac{1}{s}\bar{\psi}_s(a-x) f(x)dx \\ &= (\bar{\psi}_s \otimes f)(a) \end{aligned} \tag{6}$$

where $\otimes$ is the convolution theorem. Taking the Fourier transform and applying the convolution theorem, we can get:

$$\widehat{T^s f}(\omega) = \widehat{\bar{\psi}}_s(\omega)\widehat{f}(\omega) \tag{7}$$

where ω is a frequency component after the Fourier transform. Based on the scaling properties of the Fourier transform, $\widehat{\bar{\psi}}_s(\omega) = \widehat{\psi}^*(s\omega)$. Then by inverting the Fourier transform we can the classical wavelet transform as:

$$(T^s f)(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega x}\widehat{\psi}^*(s\omega)\widehat{f}(\omega)d\omega \tag{8}$$

By applying an impulse function, $\delta_a(x) = \delta(x-a)$, on the wavelet operator, we can localize the real-valued wavelet as

$$(T^s\delta_a)(x) = \frac{1}{s}\psi^*\left(\frac{a-x}{s}\right) = \psi_{s,a}(x) \tag{9}$$

The fundamental idea of wavelet transform is to analyze the data/signal according to a scale s at whatever location a. A vivid interpretation of wavelet analysis can be described as following: the mother wavelet acts as a window. If we look at a signal with a large "window", we would notice the gross feature, while if we take a small "window", we would notice small features. The wavelet transform can help to see both the forest (gross features) and the trees (small features) (Bruna et al., 2013).

3.4. Graph wavelet transform

Similar to graph convolution transform, graph wavelet transform also converts graph signal from vertex domain to spectral domain. The graph wavelet transform conducted on the graph Laplacian matrix is represented by a wavelet operator defined as $T_g^s = g(s\mathscr{L})$ with a kernel g at scale s, where the kernel g acts the mother wavelet in classical wavelet transform. Although the vertex domain of the graph is discrete, this kernel g is defined in the continuous domain, and thus, the scaling parameter s can be assigned as any positive real value.

The wavelet operator acts on a given function f in the spectral domain is fulfilled by modulating each Fourier mode of f as

$$\widehat{T_g^s f}(i) = g(s\lambda_i)\widehat{f}(i) \tag{10}$$

where λ_i is the i-th eigenvalue of $\mathscr{L}$. Applying inverse Fourier transform, the wavelet operator acts on f in the vertex domain can be defined as

$$(T_g^s f)(m) = \sum_{i=0}^{N-1} g(s\lambda_i)\widehat{f}(i)u_i(m) \tag{11}$$

where $m \in \{0, \cdots, N-1\}$ is an index of the m-th element/vertex in the graph. Equally, the spectral graph wavelet transform on a single vertex n can be calculated by applying the wavelet operator to a delta impulse function δ_n as

$$\psi_{s,n} = T_g^s \delta_n \# \tag{12}$$

where $\psi_{s,n} \in \mathbb{R}^N$. Similar to the graph wavelet transform performed on vertex n from any vertex m is obtained as:

$$\psi_{s,n}(m) = \sum_{i=0}^{N-1} g(s\lambda_i) u_i^*(n) u_i(m) \tag{13}$$

where u_i^* should be the conjugate eigenvector. Mathematically, based on Eq. (13), the graph wavelet coefficients of all the vertices can be written as

$$\Psi_s = U G_s U^T \# \tag{14}$$

where U is the matrix formed by Laplacian eigenvectors and $G_s = \mathrm{diag}(g(s\lambda_0), \cdots, g(s\lambda_{N-1}))$ is a diagonal kernel matrix. The graph wavelet set $\Psi_s = (\psi_{s,0}, \cdots, \psi_{s,N-1}) \in \mathbb{R}^{N \times N}$ can be considered as the graph wavelet basis. Hence, according to (Xu et al., 2019), the graph wavelet transform of a graph signal x is defined as $\hat{x} = \Psi_s^{-1} x$ and the inverse graph wavelet transform is defined as $x = \Psi_s \hat{x}$.

As stated by (Pérez-Rendón and Robles, 2004), the convolution theorem holds for continuous admissible wavelet transform that $\widehat{(\Psi_1 \otimes \Psi_2)}(\omega) = \widehat{\Psi_1}(\omega)\widehat{\Psi_2}(\omega)$, where Ψ_1 and Ψ_2 are two wavelet functions, $\otimes$ is the convolution operator, and ω is the signal in Fourier domain. Analogously, the graph convolution based on graph wavelet transform is defined as

$$f \otimes_{\mathscr{G}} x = \Psi_s((\Psi_s^{-1} f) \odot (\Psi_s^{-1} x)) \tag{15}$$

which is similar to the graph Fourier transform based graph convolution. Compare to graph Fourier transform, the graph wavelet transform based graph convolution has multiple good properties:

1. Since the structure of traffic network usually does not change, the graph wavelet matrix Ψ_s can be calculated in advance to make the training and testing process more efficient. The Chebyshev polynomial approximation of Ψ_s can make the calculation even more efficient (Xu et al., 2019).
2. The graph wavelet is well-localized in vertex domain, due to each wavelet coefficient measures the significance of the signals on a centered vertex to neighboring vertices with respect to a certain scale in the graph.
3. The wavelet coefficient matrix Ψ_s and Ψ_s^{-1} are normally highly sparse, especially for the graphs extracted from real-world traffic networks. Then, the computation can be more efficient when the sparse matrix operation is incorporated in the neural network.

3.5. Graph wavelet gated recurrent neural network

To extract the spatial-temporal features on each vertex of the graph constructed from a roadway network, we propose a graph wavelet gated recurrent (GWGR) neural network. Similar to LSTM, the GWGR has several gate units to filter out or add information to the cell state. The difference is that the gate units in GWGR are defined based on a graph wavelet matrix. The framework of the proposed model is shown in Fig. 1(d). At time t, the network-wide speed information shown in Fig. 1(b) is converted into a vector x_t as the input of the model. The traffic network structure is converted into a graph, as shown in Fig. 1(c). The graph wavelet is fixed for all time steps and it is designed based the Laplacian matrix and a kernel function.

In the proposed model, the graph wavelet coefficient matrix Ψ_s is defined in Eq. (14). We adopt the heat kernel $g(s\lambda_i) = e^{-s\lambda_i}$ and the Ψ_s^{-1} is easily obtained by replacing $g(s\lambda_i)$ with $g(-s\lambda_i)$. Then the GWGR is defined by the following equations:

$$f_t = \sigma_g(\Psi_s \Lambda_f^x \Psi_s^{-1} x_{t-1} + \Psi_s \Lambda_f^h \Psi_s^{-1} h_{t-1} + b_f) \# \tag{16}$$

$$i_t = \sigma_g(\Psi_s \Lambda_i^x \Psi_s^{-1} x_{t-1} + \Psi_s \Lambda_i^h \Psi_s^{-1} h_{t-1} + b_i) \tag{17}$$

$$o_t = \sigma_g(\Psi_s \Lambda_o^x \Psi_s^{-1} x_{t-1} + \Psi_s \Lambda_o^h \Psi_s^{-1} h_{t-1} + b_o) \tag{18}$$

$$C_t = \tanh(\Psi_s \Lambda_C^x \Psi_s^{-1} x_{t-1} + \Psi_s \Lambda_C^h \Psi_s^{-1} h_{t-1} + b_C) \tag{19}$$

where f_t, i_t, o_t and $C_t \in \mathbb{R}^N$ are the outputs of the forget gate, input gate, output gate and the input memory cell. Λ_f^x,Λ_i^x, Λ_o^x, and $\Lambda_C^x \in \mathbb{R}^{N \times N}$ are diagonal weight matrices that filter the input x_t to the three gates and the memory cell with the help of graph wavelet matrix. Similarly Λ_f^h,Λ_i^h, Λ_o^h, and $\Lambda_C^h \in \mathbb{R}^{N \times N}$are also diagonal weight matrices for the preceding hidden state h_t. b_f,b_i,b_o, and $b_C \in \mathbb{R}^N$ are four bias weight vectors. In this way, we call those matrices with the form like $\Psi_s \Lambda \Psi_s^{-1}$ as graph wavelet weight matrices in GWGR. The σ_g is the sigmoid activation function and tanh is the hyperbolic tangent function. Then the cell state C_t and the hidden state h_t at time t are calculated as follows

$$C_t = f_t \odot C_{t-1} + i_t \odot C_t \tag{20}$$

$$h_t = o_t \odot \tanh(C_t) \tag{21}$$

The $h_t \in \mathbb{R}^N$ is also the output of the GWGR unit at time t. Given the input sequence $X = [x_0, x_1, \cdots, x_{T-1}] \in \mathbb{R}^{T \times N}$, the predicted value of the future step is $\hat{x}_T = h_T$. If we only need to predict traffic data for one future step, the loss function of the model can be

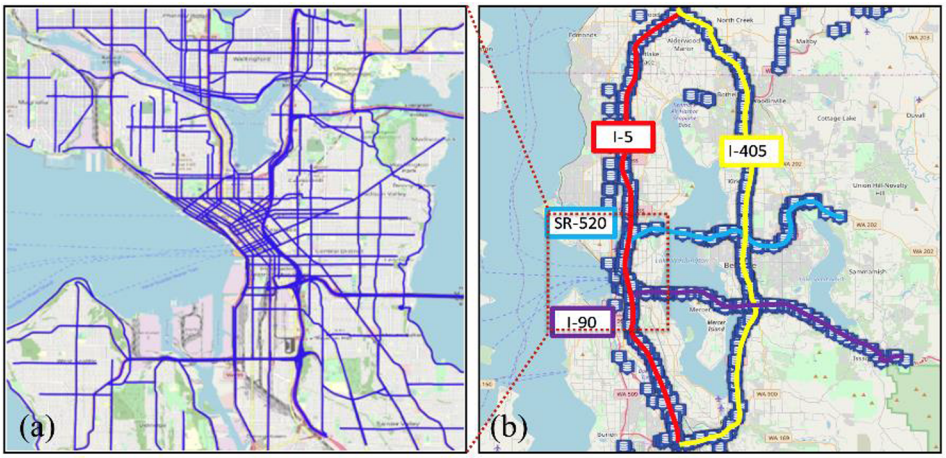


Fig. 2. (a) Urban traffic dataset covering the downtown Seattle urban corridors. (b) Freeway traffic data covering freeway system in Seattle area.

defined as

$$Loss = Loss(\hat{x}_T - x_T) = Loss(h_T - x_T) \tag{22}$$

where Loss(•) is the loss function, normally adopting the mean square error function of the traffic prediction problem.

4. Experiments

4.1. Dataset description

The proposed model is tested on two real-world datasets covering a freeway network and a more complicated urban roadway network, respectively.

Freeway Traffic Dataset (Cui et al., 2017): The data in this dataset is collected by inductive loop detectors deployed on the Seattle freeway system covering four connected freeways in the Great Seattle areas, including I-5, I-90, I-405, and SR-520, as shown in Fig. 2(b). The raw data contains three basic traffic flow characteristics, including traffic speed, volume, and density. After the dataset was comprehensively checked and cleaned (Wang et al., 2016), only the high-quality speed information in 2015 is used in the experiment. The traffic network contains 323 traffic sensing locations, i.e. $N = 323$. The time interval is 5-minute. The speed data is well-formatted and there is no missing value. This dataset is also published via an accessible link: https://github.com/zhiyongc/Seattle-Loop-Data.

Urban Traffic Dataset: This dataset originated from the National Performance Management Research Data Set (NPMRDS) data (FHWA, 2019). This dataset contains the speed data of roadway links in the Seattle downtown area, which is mostly collected by probe vehicles. In this area, the road network is very complex that it contains principal arterials, minor arterials, one-way streets, freeways, ramps, express lanes, etc. This dataset covers the year of 2012 and the time interval is also 5-minute. The roadway network contains more than 1000 roadway links, but we select the largest connected roadway network containing 745 segments in the experiment, i.e. $N = 745$, as shown in Fig. 2(a). For confidentiality reasons, this dataset is not allowed to be published at this stage.

It should be noted that the speed values of each dataset are normalized to [0,1] in the training and testing process using the following equation:

$$X = \frac{X - min(X)}{\max(X) - min(X)} \tag{23}$$

Due to this study mainly focuses on developing a new neural network structure to extract spatiotemporal features and forecast network-scale traffic status, some other traffic-related datasets, like weather data and incident data, are not incorporated in this study.

4.2. Experimental setting

Baseline Models: We compare the proposed model with the following baseline models:

(1) ARIMA (Hamed et al., 1995): the auto-regressive integrated moving average model. The model is tested by using the StatsModels 0.9.0 package (Seabold and Perktold, 2010) and the model parameters are set as default;
(2) SVR (Wu et al., 2004): the support vector regression with the radial basis function kernel. The model is tested by using the scikit-learn package (Pedregosa et al., 2011) and the model parameters are set as default;
(3) FNN: a feed-forward neural network with two hidden layers, i.e. the multi-layer perceptron model. The dimension of the hidden

layers equals the number of roadway locations/links, i.e. N;
(4) LSTM (Ma et al., 2015): a long short-term memory neural network;
(5) SGC + LSTM: a spectral graph convolution neural network stacked with an LSTM;
(6) LSGC + LSTM: a localized graph convolution neural network (Defferrard et al., 2016) stacked with an LSTM, in which the hop of graph convolution is set as 3;
(7) STGCN: Spatio-Temporal Graph Convolutional Networks, which is implemented used the source code from https://github.com/FelixOpolka/STGCN-PyTorch. The parameters of this model are kept the same as the source code in this experiment.
(8) TGCLSTM (Cui et al., 2019): a traffic graph convolution LSTM incorporating the roadway physical properties. The hop of traffic graph convolution is also set as 3.

All the baseline models are implemented or imported from existing packages using Python 3.6.8. The deep learning based models are all implemented from scratch by the authors based on Pytorch 1.0.1. These baseline models are trained and tested on a Windows 10 computer with 32 GB random-access memory (RAM) and one NVIDIA GTX 1080 Ti GPU with 11 GB memory.

Hyper-parameters: The spatial dimension N is set according to the tested datasets mentioned in Section 4.1. The temporal dimension of the input sequence is set as 10, i.e. $T = 10$. Based on the empirical tuning, the graph wavelet kernel scale s is set as 0.08 for both datasets. For both datasets, the samples are randomized and divided into training, validation, and testing sets with a ratio of 7:2:1. The batch size is set as 40 and the training loss is based on mean square error (MSE). Since the RMSProp (Tieleman and Hinton, 2012) works well for RNNs, it is used as the gradient descent optimizer whose alpha (smoothing constant) is set as 0.99 and epsilon (the term added to the denominator to improve numerical stability) is set as 10^{-8}.

Due to the tested models have different amounts of weight parameters, the best learning rates for these models are also different. In the training process, for the GWGR model, the initial learning rate for the first 10 epochs is set as 0.01 and it decreases one order of magnitude every 10 epochs after the 10th epoch. For other LSTM based models, based on the hyper-parameter tuning, the initial learning rate is set as 10^{-5}. In addition, the early stopping strategy is applied to the validation set to avoid overfitting. The training process will stop if the validation error cannot decrease 10^{-5} MSE within 10 patience steps.

Evaluation metrics: The performances of all tested models are evaluated by three metrics, including mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{24}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right| \tag{25}$$

$$RMSE = \left(\frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i|^2\right)^{1/2} \tag{26}$$

where x_i is the observed value and the $\hat{x}_i$ is the predicted value.

4.3. Experimental results

The tested prediction metrics of the urban traffic dataset and the freeway traffic dataset are shown in Table 1. The last column of Table 1 also shows the orders of magnitude of weight parameters in the tested neural networks with respect to the spatial dimension N. If the model is not based on neural network structures, such as ARIMA and SVR, the corresponding cell is intentionally left blank with a diagonal line.

The proposed GWGR outperforms all other compared models in terms of the prediction accuracy. In addition, the GWGR model requires one fewer order of magnitude of weight parameters than other models. We can notice that the deep learning based models

Table 1
Prediction performance comparison for both datasets. (The weight matrices in LSTM and GWGR has $\boldsymbol{N}^2$ and $\boldsymbol{N}$ weight parameters, respectively).

Model	Urban Traffic Dataset			Freeway Traffic Dataset			Order of Magnitude of Weight Size
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	
ARIMA	4.80	0.32	13.51%	6.10	1.09	13.85%	
SVR	4.78	0.37	13.37%	6.85	1.17	14.39%	
FNN	2.31	0.17	8.35%	4.45	0.81	10.19%	N^2
LSTM	1.14	0.09	3.88%	2.70	0.18	6.83%	N^2
SGC + LSTM	1.07	0.08	3.74%	2.64	0.12	6.52%	N^2
LSGC + LSTM	1.38	0.12	4.54%	3.16	0.23	7.51%	N^2
TGCLSTM	1.02	0.07	3.28%	2.57	0.10	6.01%	N^2
STGCN	1.34	0.09	4.33%	2.64	0.10	6.12%	N
GWGR	0.93	0.07	2.67%	2.48	0.11	5.44%	N

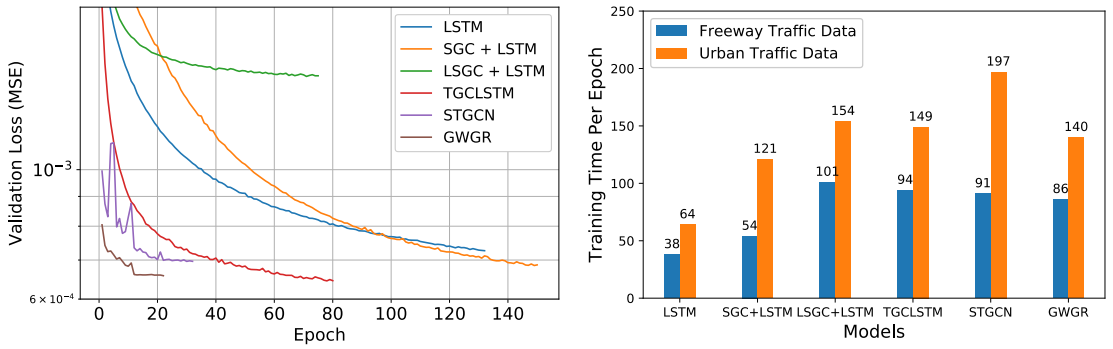


Fig. 3. (a) Validation loss versus training epoch, tested on the freeway traffic dataset. (b) Training time per epoch, tested on both datasets.

apparently work better than ARIMA and SVR for both datasets, which means the classical prediction methods do not have enough power to handle the large-scale traffic prediction problem. Among the deep learning models, the FNN does not perform as good as the other LSTM-based models. The prediction MAEs of the LSTM drop to 1.14 and 2.7 miles per hour (mph) on the urban and freeway traffic datasets, respectively, which is a pretty good performance. The SGC + LSTM and TGCLSTM have much prediction improvement on the urban traffic dataset with respect to LSTM. But the LSGC + LSTM has worse results comparing to LSTM, which may be caused by the small number of weight parameters in the localized graph convolution. The GWGR obviously achieves superior prediction results in terms of MAE and MAPE.

Although the GWGR, SGC + LSTM, and TGCLSTM all have the gated recurrent structure and take the graph adjacency matrix A as a model component, Table 1 reveals that their capabilities of capturing spatiotemporal features from a road network based graph are totally different. The GWGR apparently works better and needs fewer weight parameters. The fewer weight parameters definitely reduce the complexity of the model, and thus, greatly enhance the interpretability of the model. Further, considering the weight matrices, such as $\Psi_s \Lambda_f^x \Psi_s^{-1}$, of the GWGR are sparse, the GWGR has the great opportunity to speed up the prediction by conducting sparse matrix multiplication and to find out the key links in the traffic network by analyzing the influentialness of each graph vertex.

4.4. Training efficiency

The training efficiency of all LSTM-based models is demonstrated in this section. Fig. 3(a) shows the validation loss versus training epoch tested on the freeway traffic dataset. All the compared LSTM-based models, including LSTM, SGC + LSTM, and TGCLSTM, have N^2 magnitude of weight parameters. Comparatively, TGCLSTM converges much faster than LSTM and SGC + LSTM. The STGCN and GWGR contain less weight parameters and converge faster. During the training process, the validation loss of GWGR decrease smoother than that of STGCN. In this study, the initial learning rate of GWGR is 10^{-2} and the GWGR nearly converges after 10 training epochs, which is a really fast training process.

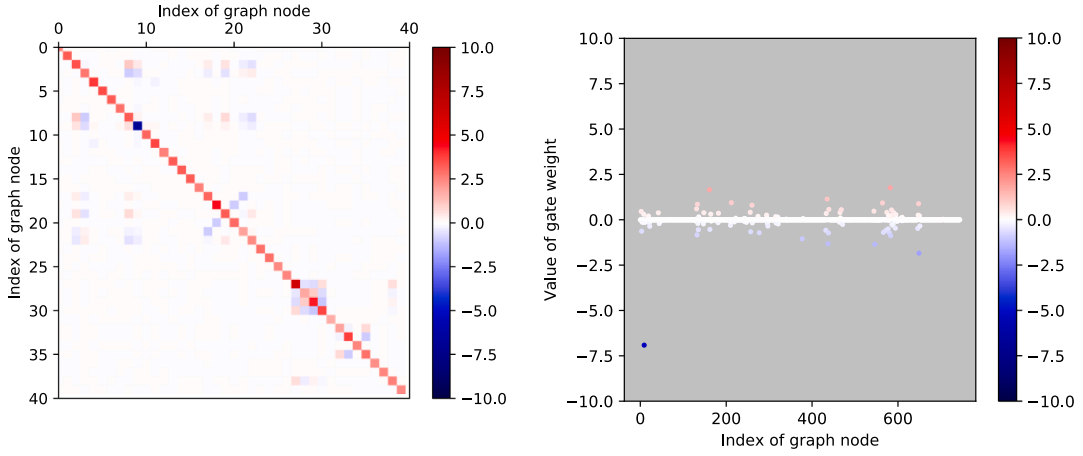
Fig. 3(b) illustrates the training time per epoch on both datasets. Since the GWGR contains more matrix multiplication operations, such as $\Psi_s \Lambda_f^x \Psi_s^{-1}$, in the gate units, the running time of GWGR definitely is more than the vanilla LSTM. SGC + LSTM's training time per epoch is less than that of GWGR, while the training times per epoch of TGCLSTM and LSGC + LSTM are even more. Besides, STGCN also take more time per training epoch than GWGR. The training time per epoch of is also Considering that GWGR needs far less training epochs, the proposed GWGR model is still the most efficient one, comparing to the LSTM-based baseline models.

4.5. Graph wavelet weight analysis

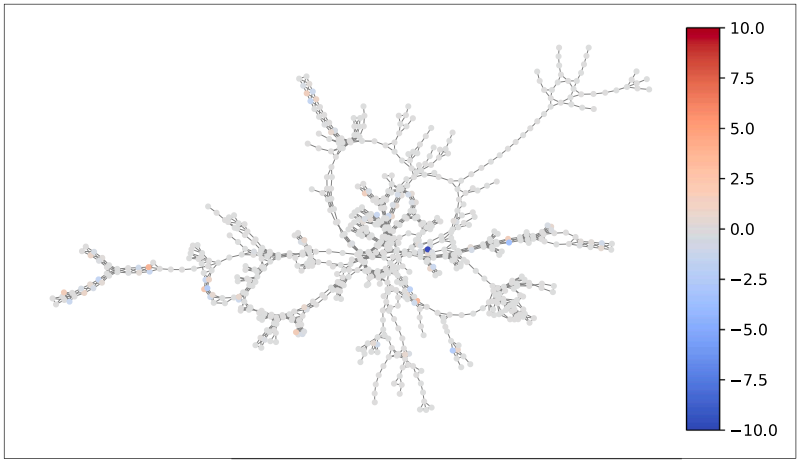
The model parameters of the GWGR consist of eight weight vectors and four bias vectors. The eight weight vectors are converted into diagonal weight matrices, such as Λ_f^x, and multiplied by graph wavelet matrices (Ψ_s and Ψ_s^{-1}) to form the graph wavelet weight matrices, such as $\Psi_s \Lambda_f^x \Psi_s^{-1}$ in the forget gate units, as shown in Eqs. (16–19). Due to the graph wavelet weight matrices are sparse and the size of the weight parameter is only $8N$ which is far fewer than N^2, the proposed GWGR is more interpretable than other LSTM based models.

In this section, the graph wavelet weight matrix of the input in the forget gate, $\Psi_s \Lambda_f^x \Psi_s^{-1}$, is selected from the GWGR tested on the urban traffic dataset and visualized to show the interpretability of the GWGR. Fig. 4(a) shows a squared section on the left top side of $\Psi_s \Lambda_f^x \Psi_s^{-1}$ containing 40 rows and 40 columns, which is sparse and symmetric. Since the $\Psi_s \Lambda_f^x \Psi_s^{-1}$ multiplies the input x_t in the forget gate, the colored pixels are the weights measuring the interactions between the speed values on different graph vertices, namely different roadway links.

When taking the 9-th row of $\Psi_s \Lambda_f^x \Psi_s^{-1}$ as an example, which contains an obvious blue dot at the 9-th column in Fig. 4(a), the columns with weight values away from zero are the ones contribute more to generate the filtered values of the 9-th graph vertex in the forget gate. Fig. 4(b) illustrates all the weight values of the 9-th row of $\Psi_s \Lambda_f^x \Psi_s^{-1}$. It can be found that only a small portion of points are away from zero. The 9-th vertex contributes the most to itself in this forget gate, since the absolute value of the weight value (the



(a) $\Psi_s \Lambda_f^x \Psi_s^{-1}$ [1:40, 1:40] (b) $\Psi_s \Lambda_f^x \Psi_s^{-1}$ [9, 1:N]



(c) Graph of $\Psi_s \Lambda_f^x \Psi_s^{-1}$ [9, 1:N]

Fig. 4. Graph wavelet weight matrix interpretation and visualization, taking the urban traffic dataset as an example. (a) Visualization of the left top part of forget gate input weight matrix $\Psi_s \Lambda_f^x \Psi_s^{-1}$, which is sparse and symmetric. (b) Scatter plot of the 9-th row of $\Psi_s \Lambda_f^x \Psi_s^{-1}$. (c) Visualization of weight on the 9-th row of $\Psi_s \Lambda_f^x \Psi_s^{-1}$ on the graph with a Kamada-Kawai layout.

dark blue dot) which is around 7.1, is the highest. Further, from Fig. 4(b), it can be distinguished that some points with higher absolute weight values are far away from the 9-th points in terms of the index of the graph vertex. However, the distance between indices of vertices might not be the actual distance between vertices in the graph.

To measure whether those graph vertices with higher absolute weight values are away from the 9-th vertex in the graph, the graph structure with a Kamada-Kawai layout is visualized in Fig. 4(c). It is easy to find a dark blue point in Fig. 4(c), which is the 9-th vertex. Then, some other light red and light blue vertices are found to be far away from the 9-th vertex in the graph. This implies the graph wavelet transform operation works well that a graph vertex is not only influenced by its neighbors but also influenced by some other vertices, which might be the key vertices in the graph or the hotspot links in the traffic network. This sparse property of the graph wavelet is quite different from the localized graph convolution, whose vertices are only influenced by fixed hops of neighboring vertices in the graph.

4.6. Graph wavelet weight matrix sparsity analysis and traffic hotspot detection

Section 4.5 presents the analysis of weight sparsity of the graph wavelet in GWGR by visualizing one of the weight matrices ($\Psi_s \Lambda_f^x \Psi_s^{-1}$) based on the urban traffic dataset. However, in this section, we quantify the sparsity of all the eight graph wavelet matrices simultaneously in order to find the most influential vertices in the graph, i.e. the most influential roadway links in the traffic network. Since the urban traffic dataset has a more complicated traffic network, it is still used for an example in this section.

Fig. 5 shows eight curves depicting how many element values in a graph wavelet weight matrix are larger than the thresholds represented by the x-axis. It can be noticed that, for all the eight matrices, only around 8% of the matrix elements are larger than 10^{-2} and only around 20% of the matrix elements are larger than 10^{-4}. Thus, it is proved that the graph wavelet is very sparse.

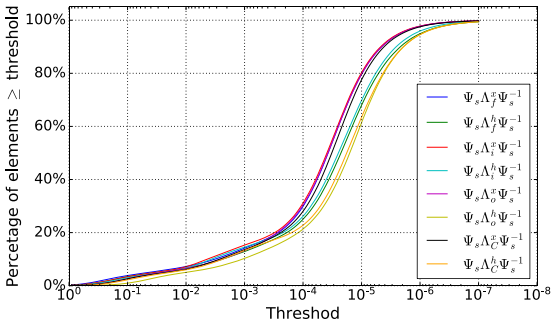


Fig. 5. Percentage of elements in the graph wavelet weight matrices that is larger than the threshold.

As tested, the values on the diagonal line of the graph wavelet weight matrices are normally larger than the non-diagonal values, which makes sense that the states of vertices tend to contribute more to themselves to generate future states. To measure the interactions between vertices and strip the vertices' self-influence out, we set all the diagonal values of the eight graph wavelet weight matrices to zero and these formatted matrices are denoted as $\Psi_s\Lambda\Psi_s^{-1*}$, where $(\Psi_s\Lambda\Psi_s^{-1*})_{ij} = 0$ if $i = j$, otherwise $(\Psi_s\Lambda\Psi_s^{-1*})_{ij} = (\Psi_s\Lambda\Psi_s^{-1})_{ij}$. Here, the diagonal weight matrix is denoted as $\Lambda \in \{\Lambda_f^x, \Lambda_i^x, \Lambda_o^x, \Lambda_C^x, \Lambda_f^h, \Lambda_i^h, \Lambda_o^h, \Lambda_C^h\}$ for simplicity.

Since these formatted matrices are all sparse, the norm functions are good indicators to show the scale/magnitude of a matrix. Here, we choose to calculate the squared $\ell 2$-norm of each formatted weight matrices, i.e. $||\Psi_s\Lambda\Psi_s^{-1*}||_2^2$, to measure the interactions between vertices. The $\ell 2$-norm of matrix $M \in \mathbb{R}^{P\times Q}$ is defined as $\sqrt{\sum_{i=1}^{P}\sum_{j=1}^{Q} M_{ij}^2}$. Intuitively, if a weight matrix has a larger squared ℓ^2-norm, it has larger absolute element values and contributes more to generate outputs.

Fig. 6(b) shows that the squared $\ell 2$-norms in the forget gate and the input gate is larger than that of the output gate and the memory cell in GWGR. This implies the forget and input gates play more important roles during the prediction process, which is mutually corroborated by (Greff et al., 2017) that the forget gate makes the LSTM-based structures more effective. Since the vertices' self-influence has been stripped out, these norm values also reveal that the graph vertices have more interactions in the forget and input gates during the prediction process.

Further, to identify the most influential vertices in the graph, we analyze the column-wise squared $\ell 2$-norms of the eight formatted weight matrices. Actually, the column-wise squared $\ell 2$-norms equals to the column-wise squared $\ell 2$-norms, since the formatted weight matrix $\Psi_s\Lambda\Psi_s^{-1*}$ is still symmetric. Mathematically, the column-wise squared $\ell 2$-norms of the k-th column of $\Psi_s\Lambda\Psi_s^{-1*}$ can be written as $\sum_{i=1}^{N}(\Psi_s\Lambda\Psi_s^{-1*})_{ik}^2$. Fig. 6(a) shows a bar chart, in which the stacked bars demonstrates the column-wise squared $\ell 2$-norms of the eight formatted weight matrices. The height of each bar indicates the sum of the squared $\ell 2$-norms of the eight formatted weight matrices at each corresponding column. In this way, the larger the total height of a bar is, the more impacts the graph vertex with the corresponding column index has on other vertices. It can be noticed that this bar chart is pretty sparse and a large portion of the sums of squared $\ell 2$-norms are close to zero. To identify the most influential links in the traffic network, we select the columns with top 5 percentage of the sums of squared $\ell 2$-norms and visualize these corresponding links on the real map, as shown in Fig. 6(c). There are 37 selected roadway links in total, which are highlighted with red color. Based on empirical investigation, these selected links are mostly located on principal arterials, intersections, freeways and freeway ramps, which are highly possible to be the hotspots in the urban traffic networks.

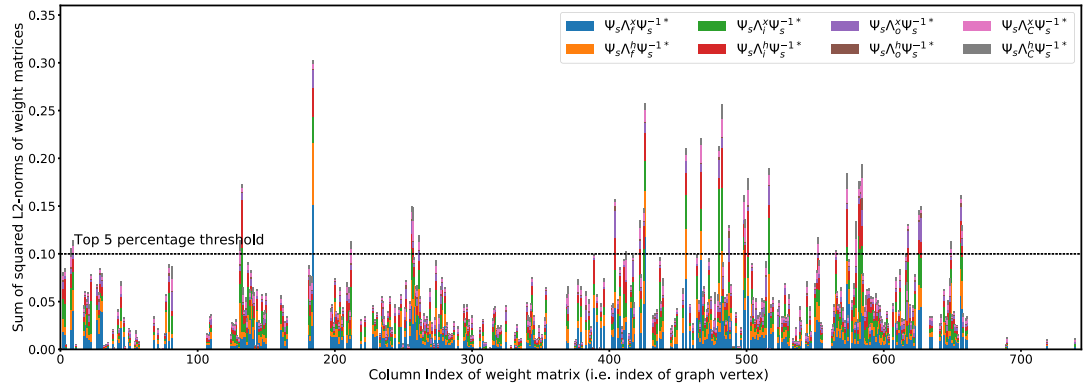
4.7. Case analysis on two tested datasets

In this section, we compare the predicted speed values with the ground truth at roadway segments selected from both the urban traffic dataset and the freeway traffic dataset, as shown in Fig. 7. All the predicted value curves and the ground truth covers fit well. Fig. 7(a) and (b) display the speed values on two different roadway links selected from the urban traffic dataset. Due to the urban area contains various types of roadways and the traffic flows are controlled by a large number of traffic lights, the urban traffic pattern is more complicated. Fig. 7(b) shows the speed values extracted from an urban corridor which are apparently smaller than the speed values extracted from a highway link, as shown in Fig. 7(a). Fig. 7(c) and (d) depict the speed values of two sensing locations selected from the freeway traffic dataset, which have different peak hours.

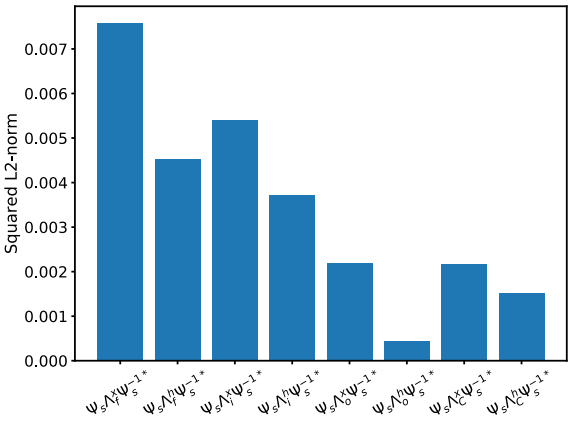
The time spans of the four sub-figures all last for one week. The difference in traffic patterns between weekdays and the weekend is also very obvious that traffic speed values fluctuate less during the weekend. In addition, we can observe that the variation of ground truth values of the urban traffic data is smaller than that of the freeway traffic data, especially at midnight, which leads the prediction error of urban traffic data is relatively smaller. In summary, as demonstrated by these figures, the proposed GWGR model has the ability to make reliable predictions simultaneously for various roadways in the traffic network.

4.8. Residual analysis

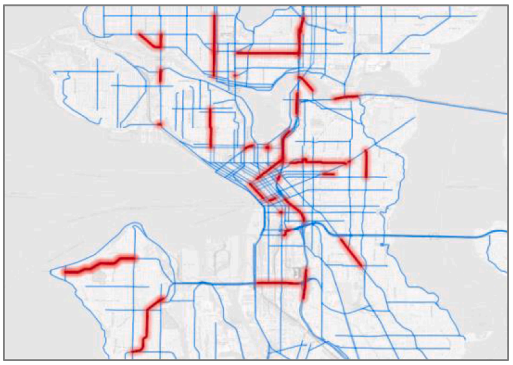
Due to residual is an important indicator to evaluate whether a model is systematically correct, the residuals of the predictions are



(a) Bar chart of the sum of column-wise squared $\ell 2$-norms of the eight graph wavelet weight matrices



(b) Squared $\ell 2$-norms of $\Psi_s \Lambda \Psi_s^{-1^*}$



(c) Visualization of urban traffic network

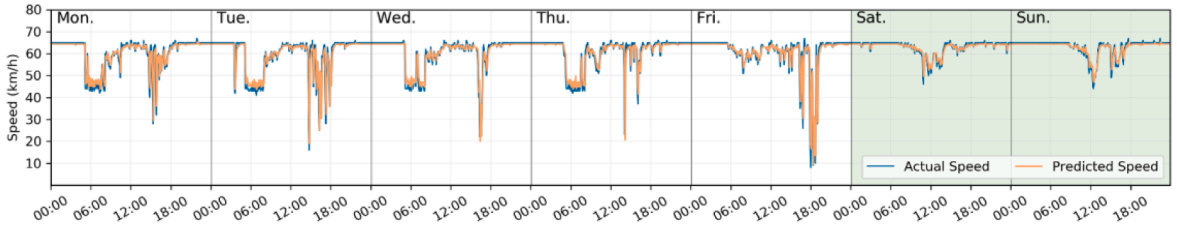
Fig. 6. (a) Bar chart of column-wise squared $\ell 2$-norms of each graph wavelet matrix. Eight column-wise squared $\ell 2$-norms calculated from the eight graph wavelet weight matrices are stacked. A horizontal dashed line shows the threshold of the top 5 percentage of sums of squared $\ell 2$-norms. (b) Squared $\ell 2$-norms of formatted graph wavelet weight matrices, i.e. $\|\Psi_s \Lambda \Psi_s^{-1^*}\|_2^2$. (c) Visualization of traffic network based on the urban traffic dataset. The roadway links having more impacts on other links (with top 5 percentage of largest column-wise squared $\ell 2$-norms) are highlighted with red color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

analyzed in this section. The residual r equals the ground truth value subtracts the predicted value, i.e. $r = x - \hat{x}$. Fig. 8(a) and (b) shows the residual distributions of the testing sets of both urban and freeway traffic datasets. Basically, the residuals of both tested datasets follow normal distributions with zero means. Although the proposed GWGR model is far more complicated than a regression model, the residual distributions with zero mean indicate the trained GWGR model has acquired sufficient predictive information.

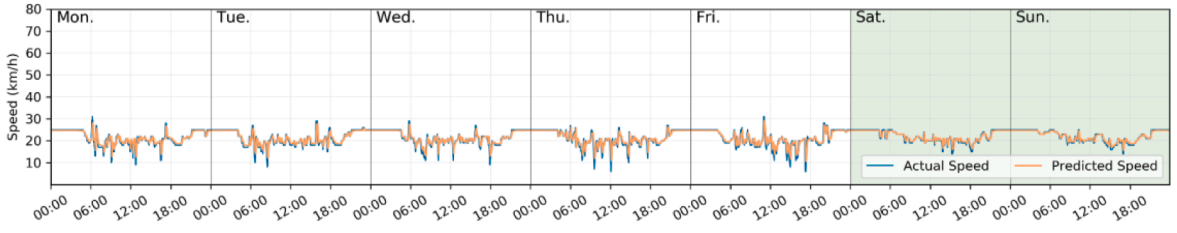
Moreover, due to traffic prediction accuracy is highly influenced by temporal information, such as the time of day, the residuals with regard to the hour of day are also evaluated by drawing boxplots, as presented in Fig. 8(c) and (d). Fig. 8(c) shows that the residuals of urban traffic data tend to be positive in the day time. It reveals that the traffic speed of roads in the urban area might have more chances to suddenly increase in the day time, taking the case shown in Fig. 7(b) as an example, leading that the predicted values are smaller than the ground truth. In the opposite, Fig. 8(d) shows that, at peak hours, the residuals of freeway traffic data tend to be negative. It also makes sense that the sudden drops of the traffic speed at rush hours will possibly result in the predicted values are larger than the ground truth. Overall, the residuals presented in Fig. 8(c) and (d) are distributed around the zero lines indicating the proposed model is capable of fitting datasets with different complexities and traffic patterns and achieving good prediction performance.

5. Conclusion

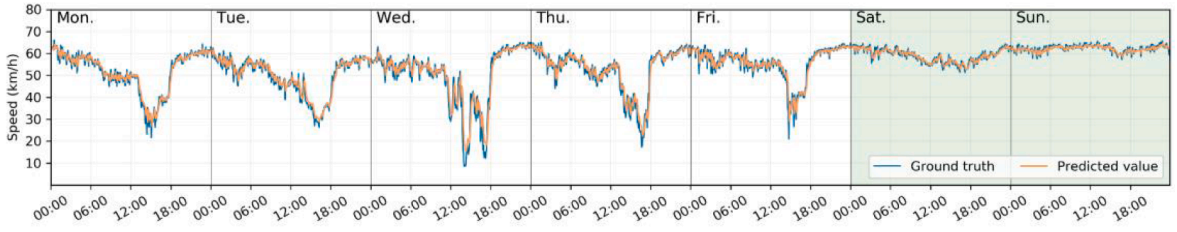
In this paper, we propose a graph wavelet gated recurrent neural network to predict network-scale traffic speed information. Since that traffic status on a road segment is highly influenced by the upstream/downstream segments and nearby bottlenecks in the traffic network, we consider the traffic network as a graph and learn flexible neighboring features from each graph node. The graph wavelet is incorporated as a key component responsible for extracting localized spatial feature. Comparing to graph convolution based model, the proposed GWGR model is more flexible, because it does not need to specify the order of hops for the vertices in the graph. To the



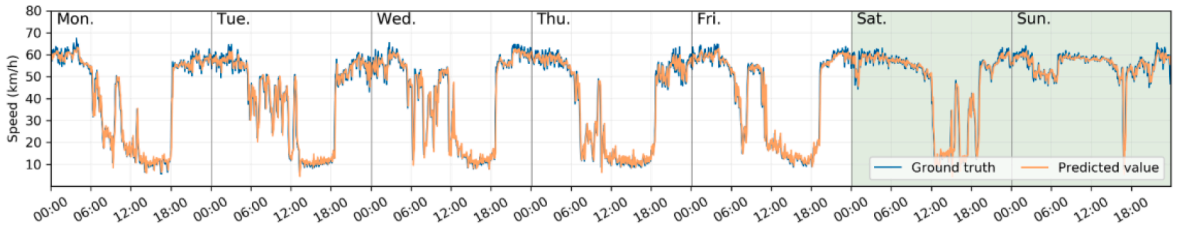
(a) Urban traffic dataset. Highway link. Road link ID: 114+04554. December 03-09, 2012.



(b) Urban traffic dataset. Urban corridor. Road link ID: 114+08028. December 03-09, 2012.



(c) Freeway traffic dataset. The sensor locates at milepost 150 of I-5, southbound. December 07-13, 2015.



(d) Freeway traffic dataset. The sensor locates at milepost 167of I-5, southbound. December 07-13, 2015.

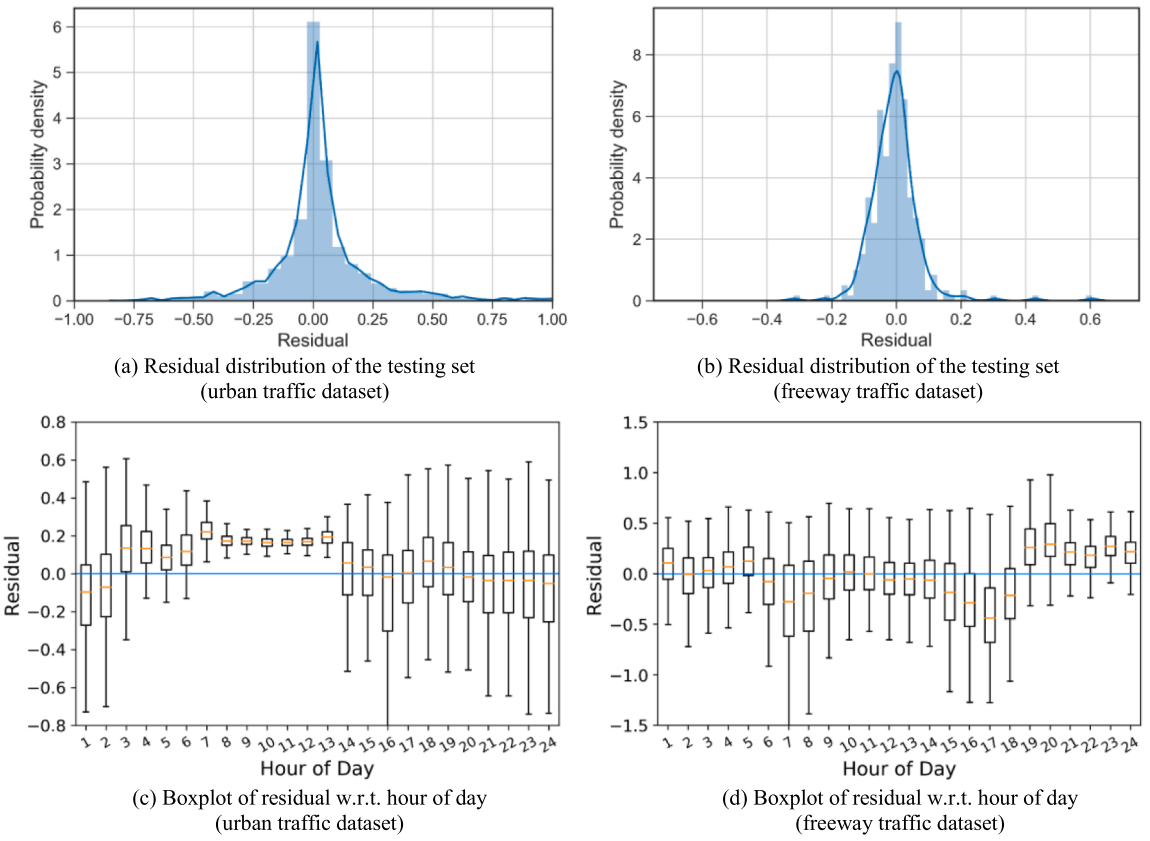
Fig. 7. Comparison of ground truth and predicted speed values tested on both datasets.

best of our knowledge, this is the first time that a graph wavelet based neural network is utilized for traffic forecasting. The GWGR model is tested on two real-world traffic datasets and the experimental results show that it can achieve superior prediction performance comparing to multiple classical and deep learning based models. Moreover, the GWGR contains less weight parameters than other LSTM-based models and achieves higher training efficiency. In addition, the model's sparse weight matrices are comprehensively analyzed and visualized. The sparsity of the weight matrices can help interpret the model and identify the most influential vertices in the traffic network based graph.

In the future, we will investigate more on the interpretation of the graph wavelet and apply graph based theories for the reasoning of traffic congestions.

CRediT authorship contribution statement

Zhiyong Cui: Conceptualization, Data curation, Methodology, Formal analysis, Validation, Writing - original draft, Writing - review & editing. **Ruimin Ke:** Conceptualization, Investigation, Software. **Ziyuan Pu:** Investigation, Data curation. **Xiaolei Ma:** Resources, Writing - review & editing. **Yinhai Wang:** Supervision, Writing - review & editing.



(a) Residual distribution of the testing set (urban traffic dataset)

(b) Residual distribution of the testing set (freeway traffic dataset)

(c) Boxplot of residual w.r.t. hour of day (urban traffic dataset)

(d) Boxplot of residual w.r.t. hour of day (freeway traffic dataset)

Fig. 8. Residual analysis plots generated based on both the urban traffic dataset and the freeway traffic dataset.

Acknowledgements

This work was supported by the Connected Cities with Smart Transportation (C2SMART) Tier 1 University Transportation Center with the USDOT Award No.: 69A3551747124. Thanks to Washington State Department of Transportation (WSDOT) for providing the research datasets. Thanks to Xinyu Chen for sharing the academic-drawing code on GitHub.

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.trc.2020.102620.

References

Bracewell, Ronald Newbold, Bracewell, Ronald N, 1986. The Fourier Transform and Its Applications. McGraw-Hill, New York.

Bruna, J., Zaremba, W., Szlam, A., LeCun, Y., 2013. Spectral Networks and Locally Connected Networks on Graphs. arXiv Prepr. arXiv: 1312.6203.

Chang, H., Lee, Y., Yoon, B., Baek, S., 2012. Dynamic near-term traffic flow prediction: system- oriented approach based on past experiences. Iet Intell. Transp. Syst. 6, 292–305.

Chen, X., He, Z., Sun, L., 2019. A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation. Transp. Res. Part C Emerg. Technol. 98, 73–84. https://doi.org/10.1016/J.TRC.2018.11.003.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv Prepr. arXiv1406.1078.

Cui, Z., Henrickson, K., Ke, R., Wang, Y., 2019. Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting. IEEE Trans. Intell. Transp. Syst.

Cui, Z., Ke, R., Wang, Y., 2017. Deep stacked bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. In: 6th International Workshop on Urban Computing (UrbComp 2017).

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering.

FHWA, 2019. National Performance Management Research Data Set (NPMRDS) [WWW Document]. URL https://ops.fhwa.dot.gov/perf_measurement/index.htm.

Ghosh, B., Basu, B., O'Mahony, M., 2009. Multivariate short-term traffic flow forecasting using time-series analysis. IEEE Trans. Intell. Transp. Syst. 10, 246–254.

Greff, K., Srivastava, R.K., Koutn\'\ik, J., Steunebrink, B.R., Schmidhuber, J., 2017. LSTM: A search space odyssey. IEEE Trans. neural networks Learn. Syst. 28, 2222–2232.

Hamed, M.M., Al-Masaeid, H.R., Said, Z.M.B., 1995. Short-term prediction of traffic volume in urban arterials. J. Transp. Eng. 121, 249–254.

Hammond, D.K., Vandergheynst, P., Gribonval, R., 2011. Wavelets on graphs via spectral graph theory. Appl. Comput. Harmon. Anal. 30, 129–150. https://doi.org/10.1016/J.ACHA.2010.04.005.

Henaff, M., Bruna, J., LeCun, Y., 2015. Deep Convolutional Networks on Graph-Structured Data. arXiv Prepr. arXiv1506.05163.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
Ke, R., Li, W., Cui, Z., Wang, Y., 2019. Two-Stream Multi-Channel Convolutional Neural Network (TM-CNN) for Multi-Lane Traffic Speed Prediction Considering Traffic Volume Impact. arXiv Prepr. arXiv1903.01678.
Li, L., Su, X., Zhang, Y., Lin, Y., Li, Z., 2015. Trend modeling for traffic time series analysis: an integrated study. IEEE Trans. Intell. Transp. Syst. 16, 3430–3439.
Li, Y., Yu, R., Shahabi, C., Liu, Y., 2017. Diffusion Convolutional Recurrent Neural Network. Data-Driven Traffic Forecasting.
Liang, Y., Cui, Z., Tian, Y., Chen, H., Wang, Y., 2018. A deep generative adversarial architecture for network-wide spatial-temporal traffic-state estimation. Transp. Res. Rec. J. Transp. Res. Board 036119811879873. https://doi.org/10.1177/0361198118798737.
Liao, B., Zhang, J., Wu, C., McIlwraith, D., Chen, T., Yang, S., Guo, Y., Wu, F., 2018. Deep sequence learning with auxiliary information for traffic prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 537–546.
Lin, L., He, Z., Peeta, S., 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: a graph convolutional neural network approach. Transp. Res. Part C Emerg. Technol. 97, 258–276.
Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.-Y., et al., 2015. Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. 16, 865–873.
Lv, Z., Xu, J., Zheng, K., Yin, H., Zhao, P., Zhou, X., 2018. LC-RNN: A Deep Learning Model for Traffic Speed Prediction. In: IJCAI 2018: 27th International Joint Conference on Artificial Intelligence. pp. 3470–3476.
Ma, X., Dai, Z., He, Z., Ma, J., Wang, Yong, Wang, Yunpeng, 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors 17, 818.
Ma, X., Li, Y., Cui, Z., Wang, Y., 2018. Forecasting Transportation Network Speed Using Deep Capsule Networks with Nested LSTM Models.
Ma, X., Tao, Z., Wang, Yinhai, Yu, H., Wang, Yunpeng, 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C Emerg. Technol. 54, 187–197. https://doi.org/10.1016/J.TRC.2015.03.014.
Okutani, I., Stephanedes, Y.J., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. Transp. Res. B – Methodol. 18, 1–11.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.
Pérez-Rendón, A.F., Robles, R., 2004. The convolution theorem for the continuous wavelet tranform. Signal Process. 84, 55–67. https://doi.org/10.1016/J.SIGPRO.2003.07.014.
Pu, Z., Liu, C., Wang, Y., Shi, X., Zhang, C., 2019. Road surface condition prediction using long short-term memory neural network based on historical data.
Seabold, S., Perktold, J., 2010. Statsmodels: Econometric and statistical modeling with python. In: Proceedings of the 9th Python in Science Conference. p. 61.
Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P., 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. 30, 83–98. https://doi.org/10.1109/MSP.2012.2235192.
Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P., 2012. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. https://doi.org/10.1109/MSP.2012.2235192.
Smith, B.L., Williams, B.M., Oswald, R.K., 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. Transp. Res. Part C Emerg. Technol. 10, 303–321.
Sun, S., Zhang, C., Yu, G., 2006. A Bayesian network approach to traffic flow forecasting. IEEE Trans. Intell. Transp. Syst. 7, 124–132.
Tieleman, T., Hinton, G., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA Neural Networks Mach. Learn. 4, 26–31.
Vlahogianni, E.I., Golias, J.C., Karlaftis, M.G., 2004. Short-term traffic forecasting: overview of objectives and methods. Transp. Rev. 24, 533–557. https://doi.org/10.1080/0144164042000195072.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2014. Short-term traffic forecasting: where we are and where we're going. Transp. Res. Part C Emerg. Technol. 43, 3–19. https://doi.org/10.1016/J.TRC.2014.01.005.
Wang, J., Chen, R., He, Z., 2019. Traffic speed prediction for urban transportation network: a path based deep learning approach. Transp. Res. Part C Emerg. Technol. 100, 372–385.
Wang, Y., Ke, R., Zhang, W., Cui, Z., 2016. Henrickson, K., 2016. Digital Roadway Interactive Visualization and Evaluation Network Applications to WSDOT Operational Data Usage. Diss. Univ. Washingt. Seattle, Washingt.
Williams, B., 2001. Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transp. Res. Rec. J. Transp. Res. Board 194–200.
Wu, C.-H., Ho, J.-M., Lee, D.-T., 2004. Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. 5, 276–281.
Wu, Y., Tan, H., Qin, L., Ran, B., Jiang, Z., 2018. A hybrid deep learning based traffic flow prediction method and its understanding. Transp. Res. Part C Emerg. Technol. 90, 166–180. https://doi.org/10.1016/J.TRC.2018.03.001.
Xu, B., Shen, H., Cao, Q., Qiu, Y., Cheng, X., 2019. Graph Wavelet Neural Network. In: International Conference on Learning Representations.
Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z., 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In: Thirty-Second AAAI Conference on Artificial Intelligence.
Yu, B., Yin, H., Zhu, Z., 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, {IJCAI-18}. International Joint Conferences on Artificial Intelligence Organization, pp. 3634–3640. https://doi.org/10.24963/ijcai.2018/505.
Yu, H., Wu, Z., Wang, S., Wang, Y., Ma, X., 2017. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors 17, 1501.
Zhang, J., Shi, X., Xie, J., Ma, H., King, I., Yeung, D.-Y., 2018. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. arXiv Prepr. arXiv1803.07294.
Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Sun, M., 2018. Graph Neural Networks: A Review of Methods and Applications.