Graph Wavelet Long Short-Term Memory Neural Network: A Novel Spatial-Temporal Network for Traffic Prediction

Qiong Wu\textsuperscript{a}, Qin Fu\textsuperscript{b} and Mingxin Nie\textsuperscript{*}

School of Wuhan University of Technology, Wuhan, China

*Corresponding author e-mail: 624968304@qq.com, \textsuperscript{a}1456889229@qq.com, \textsuperscript{b}fuqin@whut.edu.cn

Abstract. Timely accurate traffic prediction is important in the Intelligent Traffic System (ITS). It has time-varying traffic patterns and the complicated spatial dependencies on traffic network topology which makes the prediction challenging. In this paper, we propose a novel deep learning framework—Graph Wavelet Long Short-Term Memory Neural Network (GWNN-LSTM) to capture the spatial and temporal dependence simultaneously. Moreover, Graph Wavelet Neural Network (GWNN) is utilized for spatial correlations and Long Short-Term Memory Neural Network (LSTM) is used to capture the dynamic temporal correlations in traffic time series data. Experiments on real-world datasets from loop detectors in the highway of Los Angeles County (METR-LA) demonstrate that the proposed GWNN-LSTM model outperforms the state-of-the-art baselines.

1. Introduction

The goal of traffic forecasting is to predict the future traffic given historic traffic states and the underlying road networks and it is a typical problem of spatial-temporal data forecasting, which is beneficial to traffic control, traffic guidance, reduce pollution and energy saving.

Traffic forecasting has been studied for decades, falling into two main categories: parametric model and nonparametric model. Autoregressive integrated moving average (ARIMA) \cite{1} and its variants, and Kalman filtering \cite{2} are widely used in parametric models, but they usually rely on the stationarity assumption. As the parametric model cannot describe the complex non-linear and randomness of traffic flow, the non-parametric model becomes an effective solution. The common nonparametric model includes: the support vector regression model \cite{3}, the k-nearest neighbor model \cite{4}, the Bayesian network model \cite{5}, neural networks (NN) and so on, which can achieve higher prediction accuracy. However, the traditional traffic flow forecasting methods are difficult to describe the internal characteristics of the traffic data accurately.

Deep learning can learn the internal complex structure of the traffic data and make a more accurate forecasting of the traffic data. Early Deep learning models for traffic forecasting, such as deep belief networks (DBN) \cite{6} and stacked auto-encoders (SAE) \cite{7}, can achieve good forecasting performance without considering spatial and temporal dependencies jointly. Traffic data is a type of time series data, recurrent neural network (RNN) and its variants, including long short-term memory (LSTM) and gated recurrent unit (GRU) can be used to extract temporal features. Besides, convolution neural network (CNN) is proposed to model the spatial dependence of traffic condition. To fully make advantage of
spatial-temporal correlations, many researchers have combined CNNs and RNNs [8, 9, 10, 11, 12], and outperform many traditional machine learning methods. In addition, in order to select the relevant series step to make better predictions, attention mechanism is used. LC-RNN [13] uses road network embedded convolutions to learn useful spatial features, reasonably combine RNN and CNN, use CNN to learn the dynamics of neighboring areas and input them to the RNN, taking into account periodicity and additional factors, LC-RNN uses the learned parameter model to automatically fuse different information for accurate prediction. ST-ResNet [14] employ CNNs with residual connection to model the temporal closeness, period, and trend properties of crowd traffic, and the traffic data is transformed grid image as input data.

These studies above did not consider the topological structure of the traffic network. When using CNN to extract spatial features, historical traffic data was treated as pictures or grids. However, conventional CNNs are most appropriate for spatial relationships in the Euclidean space. Existing methods attempt to generalize CNNs to graph data, such as spatial methods and spectral methods. Spatial methods define convolution directly on the vertex domain, convolution is defined as a weighted average function over all vertices located in its neighborhood [15]. Another is Spectral methods define convolution via graph Fourier transform and convolution theorem, approximation by Chebyshev polynomial, it can achieve locality in spatial domain and avoiding high computational cost [16]. In recent research [17, 18, 19, 20], graph convolutional networks have been used to extract the spatial locality of k-hop neighborhood mainly via spectral methods to prediction traffic flow. How to choose the value of K and whether the localized neighbors in spatial domain truly affect the vertex is still a problem. According to the above analyses, we propose a new traffic forecasting method called GWNN-LSTM. GWNN [21] is used to extract comprehensive spatial features of traffic network, which is sparse and localized in vertex domain instead of spectral domain. And LSTM is used to capture temporal both in short-term and long term dependence.

2. Methodology
The GWNN-LSTM framework for traffic prediction is shown in Figure 1. Firstly, we use the historical time series data and traffic graph structure as input and use GWNN to learn the topological structure then extract spatial feature. Secondly, in order to capture temporal feature, we feed the time series with spatial features into LSTM to learn dynamic change on time axis. At last we get the next time step prediction result via fully connected layer. Moreover, we can use an encoder-decoder (seq2seq) architecture to make multiple step prediction.

![Figure 1. GWNN-LSTM framework for traffic forecasting.](image-url)
2.1. Traffic Prediction Problem
The goal of traffic forecasting is to predict the future traffic information (traffic speed, flow, density) based on N correlated sensors on the traffic network. We can represent the network as an undirected graph \( G = (V,E,A) \), where \( V \) is a set of nodes \( |V| = n \), \( E \) is the set of edges, and \( A \in \mathbb{R}^{n \times n} \) is the adjacency matrix. Denote the traffic data observed on \( G \) as a graph signal \( X \in \mathbb{R}^{n \times p \times T} \), where \( p \) is the number of features of each node. Let \( X(t) \) represent the graph signal observed at time \( t \), the traffic forecasting problem aims to learn a function \( f(\cdot) \) that maps \( T \) historical graph signals to the next \( h \) time steps, given a graph \( G \):

\[
[X(t+1), ..., X(t+h)] = f([X(t-T+1), ..., X(t')], G) \tag{1}
\]

2.2. Spatial Dependence Modeling
According to convolution theorem, the convolution operator on graph \( \ast \) is defined in the Fourier domain, it follows that a signal \( x \) is filtered by \( g_\theta \) as:

\[
x \ast G y = g_\theta(L)x = g_\theta(UA^T)x = U g_\theta(\Lambda)U^T x \tag{2}
\]

Where \( g_\theta(\Lambda) = \text{diag}(\theta) \) is the filter of a diagonal matrix, \( U = (u_1, u_2, ..., u_n) \in \mathbb{R}^{n \times n} \) is Laplacian eigenvectors of normalized Laplacian matrix is \( L = D^{-1/2}AD^{-1/2} \) where \( D \) is the diagonal matrix, and \( D \) is a diagonal degree matrix with \( D_{ij} = \delta_{ij} \). The Laplacian is indeed diagonalized by the Fourier basis \( L = UA^T \), where \( \Lambda = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_n) \in \mathbb{R}^{n \times n} \). The graph Fourier transform of a signal \( x \in \mathbb{R}^n \) is then defined as \( \hat{x} = U^T x \), and its inverse as \( x = U \hat{x} \). To make the filter K-localized in space and reduce its computational complexity, ChebyNet [16] restricts convolution kernel \( g_\theta \) to a polynomial expansion:

\[
g_\theta(\Lambda) = \sum_{k=0}^K \theta_k T_k(\Lambda) \tag{3}
\]

Recent researches, such as [18-22] make traffic forecasting in spectral domain via Equation (3). In this paper, we use graph wavelet transform projects graph signal from vertex domain into spectral domain via GWNN. Graph wavelet transform employs a set of wavelets as bases, defined as \( \psi_s = (\psi_{s1}, \psi_{s2}, ..., \psi_{sn}), \) where each wavelet \( \psi_{si} \) corresponds to a signal on graph diffused away from node \( i \) and \( s \) is a scaling parameter. Mathematically, \( \psi_{si} \) can be written as:

\[
\psi_{si} = U G_s U^T \tag{4}
\]

Where \( U \) is Laplacian eigenvectors, \( G_s = \text{diag}(g(s\lambda_1), g(s\lambda_2), ..., g(s\lambda_n)) \) is a scaling matrix and \( g(s\lambda) = e^{\lambda/s} \). Using graph wavelets as bases, graph wavelet transform of a signal \( x \) on graph is defined as \( \hat{x} = \psi_{s_0}^{-1}x \) and the inverse graph wavelet transform is \( x = \psi_{s_0} \hat{x} \), replacing the graph Fourier transform in Equation (2) with graph wavelet transform, we obtain the graph convolution as:

\[
x \ast G y = \psi_{s_0}(\psi_{s_0}^{-1}(x) \odot (\psi_{s_0}^{-1}(y))) = \psi_{s_0} g_\theta(\hat{G}_s) \psi_{s_0}^{-1} x \tag{5}
\]

Compared to graph Fourier transform, graph wavelet transform has the same benefits: (i) graph wavelets can be obtained via a fast algorithm; (ii) the matrix \( \psi_{s_0}^{-1} \) and \( \psi_{s_0} \) are both sparse, so graph wavelet transform is much more computationally efficient; (iii) localized in vertex domain; (iv) by varying the scaling parameter \( s \), graph wavelets are more flexible to adjust node’s neighborhoods.

According to [22], two or three layers Graph Convolutional Neural Network can achieved better result. So we use two layers GWNN in our framework to capture spatial dependence:

\[
S = \text{softmax}(g_\theta(G_s) \text{ReLU}(g_\theta(G_s)X)) \tag{6}
\]

Where \( X \in \mathbb{R}^{n \times p \times T} \) is the input data, \( S = [S(t-T+1), ..., S(t')] \in \mathbb{R}^{n \times q \times T} \).

2.3. Temporal Dependence Modeling
We leverage LSTM model the temporal dependency, which is a powerful variant of RNNs in order to void gradient exploding/vanishing. After reshaping \( S(t) \in \mathbb{R}^{n \times d} \) at each time step \( t \), the time series spatial feature \( \hat{S} \in \mathbb{R}^{n \times d \times T} \) fed into LSTM to capture temporal feature. As shown in figure 2, the forget gate \( f_t \), the input gate \( i_t \), the output gate \( o_t \) in terms of time step \( t \) are defined as follows:

\[
f_t = \delta_t(W_f \cdot \hat{S}(t) + U_f \cdot h_{t-1} + b_f) \tag{7}
\]

\[
i_t = \delta_t(W_i \cdot \hat{S}(t) + U_i \cdot h_{t-1} + b_i) \tag{8}
\]

\[
o_t = \delta_t(W_o \cdot \hat{S}(t) + U_o \cdot h_{t-1} + b_o) \tag{9}
\]

\[
h_t = o_t \cdot \sigma(W_c \cdot \hat{S}(t) + U_c \cdot h_{t-1} + b_c) + i_t \cdot (\hat{S}(t) \odot \tilde{c}_t) \tag{10}
\]

\[
\hat{c}_t = \sigma(W_c \cdot \hat{S}(t) + U_c \cdot h_{t-1} + b_c) \tag{11}
\]
\[ i_t = \delta_g (W_i \cdot S^{(t)} + U_i \cdot h_{t-1} + b_i) \] (8)
\[ o_t = \delta_g (W_o \cdot S^{(t)} + U_o \cdot h_{t-1} + b_o) \] (9)
\[ \tilde{c}_t = \tanh(W_c \cdot S^{(t)} + U_c \cdot h_{t-1} + b_c) \] (10)

Where \( W_f \), \( W_i \), \( W_o \), and \( W_c \) are the weight matrices, the \( U_f \), \( U_i \), \( U_o \) and \( U_c \) are the weight matrices connecting the previous cell output state to the three gates and the input cell state. The \( h_f \), \( b_i \), \( b_o \), and \( b_c \) are bias vectors. The \( \delta_g \) is the gate activation function, which normally is the sigmoid function, and the \( \tanh \) is the hyperbolic tangent function.

Based on the results of four above equations, at each time iteration \( t \), the cell output state \( c_t \), and the layer output, \( h_t \), can be calculated as follows:

\[ c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \] (11)
\[ h_t = o_t \cdot \tanh(c_t) \] (12)

Because it is suggested that shallow single-layer GRU and LSTM only capture short-term memories, we stack two LSTMs to extract the long-term memories and short-term memories of traffic flow. At the final time step \( t \), the hidden state \( h_t \) is fed into a fully connected to get the predicted value \( \hat{X}^{(t+1)} \). Then, the total loss function at time \( t \) can be defined as follows:

\[ \text{Loss} = L(\hat{X}^{(t+1)}, X^{(t+1)}) \] (13)

3. Research and Experiment

We conduct research on real-world large datasets: **METR-LA** [20] this traffic dataset contains traffic information, mainly traffic speed, collected from loop detectors in the highway of Los Angeles Country [23]. We select 207 sensors and collect 4 months of data ranging from Mar 1st 2012 to Jun 30th 2012 for the experiment, the total number of observed traffic data points is 6,519,002. The topology of transportation network of this traffic network is shown in Figure 3. We aggregate traffic speed readings into 5 minutes windows, and apply Z-Score normalization. 80% of data is used for training, 20% are used for testing. We predict the traffic speed of the next 5 minutes.

3.1. Experimental Settings

We compare the performance of the GWNN-LSTM model with the following baseline methods: (1) HA: Historical Average, which uses weighted average of historical data as the prediction; (2) ARIMA: Auto-Regressive Integrated Moving Average model, which is used in time series prediction.; (3) SVR: Support Vector Regression model, which uses linear support vector machine for the regression task, We use the Gaussian kernel and the penalty term is 0.01 (4) LSTM: Long Short-Term Memory Neural Network with two layers hidden, the units of the hidden layers is 64.
Figure 3. Sensor distribution of the METR-LA dataset.

All approaches based on neural network are conducted on Tensorflow, and trained using the Adam optimizer. In the experiment, we set two layers GWNN, set \( s \) to 3, set the learning rate to 0.001, the batch size to 64, and the training epoch to 1200. Table 1 shows the comparison of different approaches. These methods are evaluated based on two commonly used metrics in traffic forecasting, including:

1. Mean Absolute Error (MAE):

\[
\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|X_i - \hat{X}_i|}{X_i} \tag{14}
\]

2. Root Mean Squared Error (RMSE):

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{X}_i)^2} \tag{15}
\]

Table 1. Performance comparison of different approaches on METR-LA datasets.

| Metric | HA  | ARIMA | SVR  | LSTM | GWNN-LSTM |
|--------|-----|-------|------|------|-----------|
| MAE    | 5.15| 8.34  | 4.56 | 4.21 | 4.01      |
| RMSE   | 7.77| 9.04  | 6.78 | 5.86 | 5.50      |

Figure 4. The visualization results for prediction.

The result shows that the prediction result is more accurate than all baseline methods, and the methods bases on deep learning have powerful ability to capture the feature of traffic data. But the original data contains a large number of zero-value data, mainly because there is less traffic in the early morning. In future research, the data should be pre-processed or only daytime data should be selected for more accurate result.

4. Conclusion
In this paper, we regard the traffic prediction on traffic network as a spatiotemporal forecasting problem, and proposed the GWNN-LSTM, which can effectively captures the spatiotemporal...
dependencies. When dealing with spatial feature based on graph structure, we use GWNN which makes use of Graph wavelet transform rather than graph Fourier transform, can achieve high efficiency, localized convolution, and flexible neighborhood. When dealing with temporal feature, we use two layers LSTMs, which can capture temporal dependences both in short-term and long-term period.

Acknowledgments
This work has been supported by Project of Hubei Provincial Highway Bureau under Grant Numbers 20141h0288.

References
[1] Ahmed, Mohammed S., and Allen R. Cook. Analysis of freeway traffic time-series data by using Box-Jenkins techniques. No. 722. 1979. W. Strunk Jr., E.B. White, The Elements of Style, third ed., Macmillan, New York, 1979.
[2] kutani I, Stephanedes Y J. Dynamic prediction of traffic volume through Kalman filtering theory [J]. Transportation Research Part B: Methodological, 1984, 18 (1): 1-11.
[3] Feng X, Ling X, Zheng H, et al. Adaptive Multi-Kernel SVM With Spatial–Temporal Correlation for Short-Term Traffic Flow Prediction [J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 20 (6): 2001-2013.
[4] ZHANG X, HE G, LU H. Short-term traffic flow forecasting based on K-nearest neighbors non-parametric regression [J]. Journal of Systems Engineering, 2009, 24 (2): 178-183.
[5] Zheng W, Lee D H, Shi Q. Short-term freeway traffic flow prediction: Bayesian combined neural network approach [J]. Journal of transportation engineering, 2006, 132 (2): 114-121.
[6] Huang W, Song G, Hong H, et al. Deep architecture for traffic flow prediction: deep belief networks with multitask learning [J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15 (5): 2191-2201.
[7] Lv Y, Duan Y, Kang W, et al. Traffic flow prediction with big data: a deep learning approach [J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 16 (2): 865-873.
[8] Wu Y, Tan H. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework [J]. arXiv preprint arXiv: 1612.01022, 2016.
[9] Wu Y, Tan H, Qin L, et al. A hybrid deep learning based traffic flow prediction method and its understanding [J]. Transportation Research Part C: Emerging Technologies, 2018, 90: 166-180.
[10] Du S, Li T, Gong X, et al. A hybrid method for traffic flow forecasting using multimodal deep learning [J]. arXiv preprint arXiv: 1803.02099, 2018.
[11] Yao H, Tang X, Wei H, et al. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction [C] // AAAI Conference on Artificial Intelligence. 2019.
[12] Yao H, Wu F, Ke J, et al. Deep multi-view spatial-temporal network for taxi demand prediction [C] // Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
[13] Lv, Z., Xu, J., Zheng, K., Yin, H., Zhao, P., & Zhou, X. (2018, January). LC-RNN: A Deep Learning Model for Traffic Speed Prediction. In IJCAI (pp. 3470-3476).
[14] Zhang J, Zheng Y, Qi D, et al. Predicting citywide crowd flows using deep spatio-temporal residual networks [J]. Artificial Intelligence, 2018, 259: 147-166.
[15] Monti F, Boscai D, Masci J, et al. Geometric deep learning on graphs and manifolds using mixture model cnns [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 5115-5124.
[16] Defferrard M, Bresson X, Vanderghynst P. Convov-lutional neural networks on graphs with fast localized spectral filtering [C] // Advances in neural information processing systems. 2016: 3844-3852.
[17] Cui Z, Henrickson K, Ke R, et al. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting [J]. arXiv preprint
arXiv: 1802.07007, 2018.

[18] Diao Z, Wang X, Zhang D, et al. Dynamic Spatial-Temporal Graph Convolutional Neural Networks for Traffic Forecasting [J]. 2019.

[19] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting [J]. arXiv preprint arXiv: 1709.04875, 2017.

[20] Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting [J]. arXiv preprint arXiv: 1707.01926, 2017.

[21] Xu B, Shen H, Cao Q, et al. Graph Wavelet Neural Network [J]. arXiv preprint arXiv: 1904.07785, 2019.

[22] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [J]. arXiv preprint arXiv: 1609.02907, 2016.

[23] Jagadish H V, Gehrke J, Labrinidis A, et al. Big data and its technical challenges [J]. Communications of the ACM, 2014, 57 (7): 86-94.