Attention based graph Bi-LSTM networks for traffic forecasting

Zhao, Han; Yang, Huan; Wang, Yu; Wang, Danwei; Su, Rong

2020

Zhao, H., Yang, H., Wang, Y., Wang, D., & Su, R. (2020). Attention based graph Bi-LSTM networks for traffic forecasting. Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC). doi:10.1109/ITSC45102.2020.9294470

<https://hdl.handle.net/10356/146155>

<https://doi.org/10.1109/ITSC45102.2020.9294470>

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at:
<https://doi.org/10.1109/ITSC45102.2020.9294470>

Downloaded on 26 Apr 2023 17:57:57 SGT

Attention Based Graph Bi-LSTM Networks for Traffic Forecasting

Han Zhao¹, Huan Yang¹, Yu Wang¹, Danwei Wang¹ and Rong Su¹

Abstract—Traffic forecasting is of great importance to vehicle routing, traffic signal control and urban planning. However, traffic forecasting task is challenging due to several factors, such as complex spatial topological structure and dynamic changing of traffic status. Most existing methods have limited ability to capture both spatial and temporal dependence of traffic data. In this paper, we propose a novel end-to-end deep learning model, Attention based Graph Bi-LSTM networks (AGBN) to perform the traffic forecasting task. It uses graph convolutional network (GCN) to extract spatial features and bidirectional long short-term memory networks (Bi-LSTM) to capture the temporal dependence. The attention mechanism is used to select relevant features at all time steps. Experiments show that our model could extract both spatial and temporal dependence well and outperforms other baselines on real-world traffic datasets.

I. INTRODUCTION

Advanced intelligent transportation system is inseparable from the processing of basic traffic data. Traffic flow prediction is a significant branch of traffic data processing. Accurate traffic flow prediction information can provide traffic managers with a strong basis for traffic decision-making, let drivers choose a smooth road trip to avoid or alleviate traffic congestion. In traffic light control system, a good prediction of traffic flow could help adjust the phase time and optimize the total delay of intersections.

Previously, traffic data are mainly collected in a fixed collection technology [1]. The sensors are generally fixed at the main road, intersections and highway entrance. With the development of mobile collection technology, some dynamic traffic data could be captured, such as GPS (Global Positioning System) data. These traffic data are in multi-dimensional and contain both spatial and temporal information. How to effectively apply these data to carry out the prediction task has become a research hot spot. However, traffic forecasting is always a highly challenging task due to the following impacting factors:

(1) Complex spatial topology. The topological structure of the urban road network is extremely complex. It is difficult to model and extract the spatial features from road networks. However, the spatial dependence between intersections is not neglectable. The adjacency roads may have strong relations as the traffic flow transfer effect. For instance, the traffic flow at upstream of one intersection would affect the downstream strongly [2].

(2) Dynamic traffic status. The traffic status is changing dynamically over time due to traffic signals and accidents on the road. It is hard to capture the dynamic temporal correlations between each time step. However, traffic status

is periodic over a day. Hence, how to combine the periodic pattern with dynamic temporal dependence has risen attention of researchers.

Recently, as the development of machine learning algorithms, some deep learning methods are applied into various traffic prediction tasks. For instance, Huang et al. [3] proposed a deep belief networks (DBN) in traffic forecasting, which first applies deep learning to transportation research. However, the dense network fails to extract both temporal and complex spatial features from the traffic input data. Some researchers also apply Convolutional neural network (CNN) to capture spatial dependence in traffic forecasting tasks [4]. However, CNN only works well on Euclidean data, such as images, audio, etc. For the complex topological structure of road network, the traditional convolution operation could rarely extract the spatial features from the road network. At the temporal side, Long short-term memory network (LSTM) [5] is well applied to perform the time series prediction task. The gating mechanism could facilitate the regulation of long-term memory. However, the performance of the memory network would deteriorate as the time length increases [6]. Moreover, a single LSTM network could only capture the temporal features.

In order to solve the above challenges, we proposed a novel end-to-end deep learning model: Attention Based Graph Bi-LSTM Networks (AGBN) to perform the traffic forecasting task. Our model combines a graph convolutional network (GCN) [7] with attention based bi-directional LSTM. The road network is modelled as a graph and GCN performs convolution operation on graphs, which capture the spatial dependence efficiently. The attention mechanism has been successfully applied in image recognition and machine translation [8] [9], which could efficiently prevent the loss of information. Hence, the attention based bi-directional LSTM could capture the temporal dependence and select the most important features from each time step. We evaluate our model on several real-world traffic datasets. The experiments show that our model excels at all metrics compared to other baselines.

The remainder of paper is organized as follows: Section II reviews the studies on traffic prediction. Section III introduces the details of our method. IV demonstrates the experiment results on real-world traffic datasets. Section V concludes the paper.

II. RELATED WORK

In recent decades, experts around the world have developed a variety of prediction models for short-term traffic flow prediction by using the methods of various disciplines. The

¹School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. E-mail: zhao0278@e.ntu.edu.sg

major methods could be divided into three types: statistic-based methods, non-parametric model and deep learning-based methods.

Statistics-based model. Statistics-based models include historical average model [10], autoregressive integrated moving average model [11] and Kalman filtering model [12], etc. Historical Average method considers traffic flow as a seasonal process, and uses weighted average of historical seasons to predict current value. The main issues is that historical average method could not capture small changes in forecasting horizon because it does not depend on short-term data. Autoregressive integrated moving average (ARIMA) is another popular model for traffic forecasting problem. In 1995, Hamed et al. [11] used the ARIMA model to predict the traffic volume in urban arterial. Although the ARIMA model is said to be optimal forecast in the statistical field, it is only valid for those models which are linear combinations of historical time steps with fixed coefficients.

Non-parametric model. Due to the non-linearity of traffic data, non-parametric models received more attention than statistics-based model. Non-parametric models are easy to build and could interpret the non-linearity in a good way. For instance, Support vector regression (SVR) is generally used in traffic forecasting task [13]. The traffic data are mapped a feature space and a linear regression is performed within that space. However, it has been proved that SVR has limitations in high-dimensional and complex traffic state due to the shallow architecture of network [14]. Besides, K-nearest neighbor method [15] is also a widely used non-parametric regression method to forecasting short-term traffic flow, and has proved to be accurate and portable potential by extensive prediction results. Nevertheless, these models are still not able to deal with complex road network and dynamic traffic status.

Deep learning-based model. Nowadays, it has become an inevitable trend to analyze and predict big traffic datasets through deep learning methods. The deep learning-based model has achieved a huge success in several fields, such as convolutional neural network (CNN) [16] in computer vision and recurrent neural network (RNN) in natural language processing. Many researchers [17] [18] integrate CNN and RNN into one spatial-temporal model to capture the features of past traffic flow. These methods utilizes the temporal and spatial characteristics of traffic flow to perform the forecasting task. Normally, the road network is segmented to regular 2-D grid data and convolution operation is applied to extract the spatial features. Although these methods made a great success in traffic forecasting tasks, it has certain defects on the capture of spatial features. The CNN model could only extract some simple spatial features on the regular road networks. For complex urban road network, the adjacency relationship between intersections could not be interpreted in these models.

Motivated by these studies, we proposed our novel model AGBN to solve the challenges and issues above. Section III would explain the details of our model.

III. METHODOLOGY

A. Problem Description

The structure of road network in this paper could be described as a weighted directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, A\}$, where $|\mathcal{V}| = N$ is a set of nodes (e.g., sensors points); $\mathcal{E}$ is a set of edges connecting the vertices, which represents the connectivity among them; $A \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix of graph $\mathcal{G}$, which represents the proximity between every two vertex.

The feature matrix $X_t \in \mathbb{R}^{N \times C}$ denotes the traffic flow observed at time t on the graph $\mathcal{G}$, where C is the dimension of traffic flow data (e.g., traffic volume, speed, occupancy, etc.). Hence, the traffic forecasting problem could be stated as follow: Given the observations of historical P time steps $\mathcal{X}_t = \{X_{t-P}, X_{t-P+1}, \dots, X_t\}$ and graph $\mathcal{G}$, predict the future traffic flow at next Q time steps $\hat{\mathcal{Y}}_t = \{\hat{Y}_{t+1}, \dots, \hat{Y}_{t+Q}\}$, where $\hat{Y}_{t+T}$ denotes the estimated future feature matrix at time $t + T$. It aims to learn a function f which maps historical observation data to future traffic flow:

$$\hat{\mathcal{Y}}_t = f(\mathcal{G}, \mathcal{X}_t) \quad (1)$$

B. Structure of Network

In the remainder of this section, the network architecture of AGBN would be elaborated. As shown in Figure 1, AGBN is composed of several layers. The input traffic data are fed into the graph convolutional layer, which extract the spatial dependence from input traffic graph. The spatial output will be considered as the input of temporal layer. The Bi-LSTM layer with attention mechanism would capture the temporal dependence of each time step. The most essential features from each time step would be selected. Finally, with a fully connected (Dense) layer, the output is mapped to the prediction results, which contains the next Q steps of traffic flow.

Fig. 1. Architecture of AGBN model for traffic forecasting

- **Spatial dependency modelling.** The topological structure of urban road network is complex. Hence, how to extract the spatial dependency from the complex structure becomes a crucial problem. Generally, Convolutional Neural Network (CNN) is used to extract the spatial features. However, it can be only applied to Euclidean data, such as images, audio frames, etc. These types of data are in the form of regular grids. Urban road network is considered as a type of Non-Euclidean data, which could not be represented in a single 2D or 3D matrix. Thus, the convolution operation is no longer applicable to non-Euclidean data. Recently, the convolutional neural network has been generalized on graph, which could perform the convolution operation on graph data in spectral domain. The Graph Convolutional Network (GCN) could take arbitrary graph structured data as input. It shows a good ability to capture spatial dependency and aggregate features among nodes. The spectral convolution operation could be written as:

$$\begin{aligned} g_\theta *_{\mathcal{G}} x &= g_\theta(L)x \\ &= g_\theta(U\Lambda U^T)x \\ &= Ug_\theta(\Lambda)U^T x \end{aligned} \quad (2)$$

In Eq. (2), g_θ is a convolutional kernel and “ $*_{\mathcal{G}}$ ” denotes the convolutional operator. The normalized graph Laplacian $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is eigendecomposed (D is the diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$), where U is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues. However, the computation cost of eigendecomposition is expensive when the graph is at large scale. To solve this problem, Chebyshev polynomials technique are used to approximate the convolution operation [19],

$$g_\theta *_{\mathcal{G}} x = g_\theta(L)x \approx \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x \quad (3)$$

where $T_k(\tilde{L})$ is the Chebyshev polynomial of order k and the graph Laplacian is scaled to $\tilde{L} = 2L/\lambda_{\max} - I_n$ [20]. For further simplification, a 1st-order approximation is applied, which simplifies Eq. (3) to,

$$\begin{aligned} g_\theta *_{\mathcal{G}} x &\approx \theta_0 x + \theta_1 \left(\frac{2}{\lambda_{\max}} L - I_n \right) x \\ &\approx \theta_0 x - \theta_1 \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) x \end{aligned} \quad (4)$$

By replacing θ_0 and θ_1 to a single parameter θ , where $\theta_0 = -\theta_1 = \theta$. The final graph convolution operator could be expressed as,

$$\begin{aligned} g_\theta *_{\mathcal{G}} x &= \theta \left(I_n + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) x \\ &= \theta \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \right) x \end{aligned} \quad (5)$$

where $\tilde{A} = A + I_n$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. Thus, the convolutional layer could be expressed as follow,

$$M_t = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \mathcal{X}_t \theta) \quad (6)$$

where σ is an activation function which reflect the non-linearity of the model. M_t would be the output spatial sequence in each time step. Each node will aggregate the feature from neighbors and update themselves respectively.

- **Temporal dependence modelling.** After the graph convolutional layer, the spatial dependence has been captured from the neighbors of each node in the graph. Extracting the temporal dependence is also a key problem in this traffic forecasting problem. The long short-term memory (LSTM) model could capture the long-term temporal dependencies of the traffic time series, where the gated mechanism is used to memorize the long-term information. The attention mechanism is also applied to the temporal dependence model, which would calculate the importance of each time step and allocate a weight to each of them. The LSTM model aims to learn a mapping function from M_t to $\mathbf{h}_t$,

$$\mathbf{h}_t = LSTM(\mathbf{h}_{t-1}, M_t) \quad (7)$$

where $\mathbf{h}_t$ is the hidden state at time t . Each LSTM unit has a memory cell with state $\mathbf{c}_t$. Three gates are implemented to access the memory cell: forget gates $\mathbf{f}_t$, input gate $\mathbf{i}_t$ and output gate $\mathbf{o}_t$. The following equations demonstrate the gated mechanism of LSTM,

$$\mathbf{f}_t = \sigma(\mathbf{W}_f [\mathbf{h}_{t-1}; M_t] + \mathbf{b}_f) \quad (8)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i [\mathbf{h}_{t-1}; M_t] + \mathbf{b}_i) \quad (9)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o [\mathbf{h}_{t-1}; M_t] + \mathbf{b}_o) \quad (10)$$

The current cell state $\mathbf{c}_t$ would be updated by both previous cell state $\mathbf{c}_{t-1}$ and current information generated by the cells.

$$\begin{aligned} \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} \\ &+ \mathbf{i}_t \odot \tanh(\mathbf{W}_c [\mathbf{h}_{t-1}; M_t] + \mathbf{b}_c) \end{aligned} \quad (11)$$

where $\odot$ denotes the Hadamard product, and

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (12)$$

In this paper, a bidirectional LSTM [21] is applied to capture the temporal dependence. The network contains a forward propagation which processes spatial sequence from M_{t-P+1} to M_t , but also a back propagation from M_t to M_{t-P+1} . Hence, the hidden outputs of two directions are concatenated as the final hidden state of this bidirectional LSTM model.

$$\mathbf{h}'_t = \left[\vec{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t \right] \quad (13)$$

With the computed hidden states $\mathbf{h}'_t$, the temporal attention weights are calculated as follow:

$$\mathbf{e}_t = \tanh(\mathbf{W}_e \mathbf{h}'_t + \mathbf{b}_e) \quad (14)$$

$$\alpha_t^i = \frac{\exp(\mathbf{e}_t^i)}{\sum_{j=1}^P \exp(\mathbf{e}_t^j)} \quad (15)$$

The attention weight α_t^i represents the i -th hidden state for the prediction. The attention mechanism would compute a output vector $\mathbf{s}_t$ as a weighted sum of all hidden states.

$$\mathbf{s}_t = \sum_{i=1}^P \alpha_t^i \mathbf{h}'_i \quad (16)$$

Finally, the output $\mathbf{s}_t$ would be mapped to the predicted sequence $\hat{\mathcal{Y}}_t = \{\hat{Y}_{t+1}, \dots, \hat{Y}_{t+Q}\}$ by a fully connected layer,

$$\hat{\mathcal{Y}}_t = \sigma(\mathbf{W}_s \mathbf{s}_t + \mathbf{b}_s) \quad (17)$$

where $\mathbf{W}_s$ is the weighted matrix and $\mathbf{b}_s$ is the bias at the output layer.

C. Training Procedure

The stochastic gradient descent (SGD) method with Adam optimizer is used to train the model. The parameters could be learned by the backpropagation with mean square error as the objective function:

$$\mathcal{L}(\Theta) = \frac{1}{n} \sum_{t=1}^n (\mathcal{Y}_t - \hat{\mathcal{Y}}_t)^2 \quad (18)$$

where $\mathcal{Y}_t$ is the ground truth of traffic flow, $\hat{\mathcal{Y}}_t$ is the estimated traffic flow, n is the number of training samples and Θ is the parameters to be optimized.

IV. EXPERIMENTS

A. Dataset Description

We evaluate the performance of our model AGBN on two freeway traffic datasets: PeMS04 and PeMS07. These data are collected from the Caltrans Performance Measurement System (PeMS) in each 30 seconds [22]. The dataset contains three types of feature: traffic volume, traffic speed and occupancy rate. The traffic flow data are aggregated at each 5 minutes. For PeMS04, 182 sensor points are selected after removing the redundant detectors. The time range is from Jul to Aug in 2017. In PeMS07, 235 sensors are selected and the time range is from April to May in 2016.

B. Data Preprocessing

The data preprocessing is necessary before the raw data are input to the model. Due to the malfunction of detector sensors and data transmission failure, some of the data are missing. To solve this problem, a linear interpolation method is used to fill the missing values. Also there are some outliers in the PeMS freeway data. Considering the speed limit on the highway, a maximum threshold of 200 km/h

is set to remove those outliers. In addition, some redundant detectors are removed to ensure that the distance between any two detectors is not too far away. After that, the traffic flow data are normalized into the range (0,1) with min-max normalization.

$$x^* = \frac{x - \min}{\max - \min} \quad (19)$$

The weighted adjacency matrix is computed based on the road distance between sensors. A threshold ϵ is set to ensure the sparsity of the adjacency matrix [23],

$$A_{i,j} = \begin{cases} \exp\left(-\frac{d_{i,j}^2}{\sigma^2}\right), & \text{if } \exp\left(-\frac{d_{i,j}^2}{\sigma^2}\right) \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

where $d_{i,j}$ is the distance between detector i and j . σ is the standard deviation.

C. Metrics

The following metrics are used to evaluate the prediction performance of AGBN.

- Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (\mathcal{Y}_t - \hat{\mathcal{Y}}_t)^2} \quad (21)$$

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{t=1}^n |\mathcal{Y}_t - \hat{\mathcal{Y}}_t| \quad (22)$$

- Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{\mathcal{Y}_t - \hat{\mathcal{Y}}_t}{\mathcal{Y}_t} \right| \quad (23)$$

D. Training Parameters

In our experiments, all the datasets are split into 70% training dataset and 30% test dataset. The prediction time horizon is set to next 15 minutes, 30 minutes and 45 minutes respectively. The learning rate is set to 0.001 and the batch size is set to 32. Some dropout and L2 regularization are used to avoid the overfitting problem.

E. Baseline Model

The following baseline models are used to compare the performance of the AGBN model:

- (1) **Autoregressive Integrated Moving Average model (ARIMA)**. It is a statistical model characterized by Auto Regressive (AR), Moving Average (MA) and order of differencing.
- (2) **Support Vector Regression model (SVR)**. It maps multi-dimensional traffic data into a feature space and performs a linear regression within the space.
- (3) **Multi-layer Perceptron (MLP)**. It consists of two hidden layers, which are fully connected in a feed-forward way.
- (4) **Long Short-Term Memory network (LSTM)**. The time sequence is directly input to this model without spatial dependence modelling.

TABLE I
TRAFFIC PREDICTION PERFORMANCE COMPARISON BY THREE EVALUATION METRICS.

Dataset	Method	15 min			30 min			45 min		
		RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)
PeMS04	ARIMA	8.09	6.13	13.65	8.16	6.28	14.62	8.31	6.52	16.21
	SVR	7.24	4.65	10.12	7.45	4.77	10.46	7.66	4.92	10.80
	MLP	7.30	4.72	10.41	7.61	4.86	10.98	7.78	5.03	11.20
	LSTM	6.89	4.33	9.86	6.99	4.51	10.11	7.10	4.73	10.35
	AGBN	6.65	4.20	9.26	6.74	4.35	9.40	6.86	4.54	9.55
PeMS07	ARIMA	8.23	6.32	14.32	8.41	6.47	16.19	8.55	6.73	18.33
	SVR	7.52	5.12	11.34	7.63	5.22	11.57	7.78	5.37	11.78
	MLP	7.67	5.31	12.65	7.84	5.45	13.02	8.02	5.81	13.54
	LSTM	7.02	4.67	10.23	7.11	4.77	10.44	7.23	4.82	10.56
	AGBN	6.83	4.32	9.83	6.91	4.45	9.97	7.02	4.61	10.12

F. Comparison and Result Analysis

Fig. 2. Visualization results on PeMS04 in one day

Table I demonstrates the evaluation result of AGBN compared to several baseline models on the dataset PeMS04 and PeMS07. We could observe that statistic-based model ARIMA performs worse in all time horizon, especially at 45 min. It is because that the PeMS traffic data are in high dimension and the road network structure is complex. Hence, the statistic-based model could not interpret the non-linearity, let alone the spatial dependence in the road network. The deep learning-based model generally performs better than the traditional machine learning models. It can be observed that due to the long-term memory gating mechanism, LSTM performs much better than other models at long time forecasting tasks. Our model takes the advantage of spatial and temporal dependence modelling, which achieved the best results in all time horizons and evaluation metrics.

Figure 2 demonstrates the predicted results of AGBN over other baselines in one day range. It could be observed that AGBN could capture the dynamic change of traffic status from 17:00 to 20:00. For the response time of congestion prediction, the ARIMA model and historical average model both have some delay time to capture the congestion status. However, our AGBN model could aggregate the traffic status from other adjacency junctions, where the traffic features at other detector points would have a strong affect on the current detector point. Hence, AGBN model could predict the congestion status in a quick time. Figure 3 demonstrates the scatter plot between real speed and prediction speed. It could be observed that AGBN model fits the straight line

Fig. 3. Scatter plot of speed prediction on PeMS04

better than other models. For Historical Average model, it only takes the average of historical values. Hence, at low speed band, the prediction performance is not good. For ARIMA model, the performance result is the worst, which overestimated the traffic speed at congestion time period. For further analysis, we would focus on the significance of spatial dependence modelling and long term information capture.

Significance of spatial dependence modelling. To verify that AGBN could extract the spatial features from the complex topology structure, we focus on the comparison between our model and LSTM only. As shown in table I, it is obviously that our model has a better performance at all prediction time steps. Compared to the LSTM model at 30-min and 45-min prediction task, the RMSE of our model is decreased by 3.58 % and 3.38% respectively, which indicates that our model could capture the temporal dependence efficiently.

Analysis of long-term forecasting. The long-term prediction is more challenging. As the prediction time horizon increases, the performance of all models will inevitable

Fig. 4. Visualization results on PeMS07 - 30 minutes horizon

Fig. 5. Visualization results on PeMS07 - 45 minutes horizon

decrease. As shown in Figure 4 and 5, the error between ground truth and predicted value becomes higher when the time horizon increases. However, regardless of the change of horizon, our model still achieves the best performance in long-term prediction. It verifies that the long-term memory gating mechanism is highly effective. Furthermore, attention mechanism successfully selected the most important features in the traffic time sequence. For the periodic traffic pattern, our model could accurately predict the trend of low-speed time duration, which may forewarning the traffic congestion in dozens of minutes advance.

V. CONCLUSIONS

In this paper, a novel end-to-end deep learning model is proposed for traffic forecasting task, which combines graph convolutional network and bi-directional long short-term memory network with attention mechanism. The complex topology spatial features and dynamic temporal features are well extracted from the spatial-temporal traffic data. The experiments on the real-world traffic datasets show that our model has some improvement compared to several baseline models. In future work, more exogenous factors may be considered into our model, such as weather condition and traffic signals.

REFERENCES

- [1] W. Wang and G. Wets, *Computational intelligence for traffic and mobility*. Springer Science & Business Media, 2012.
- [2] C.-j. Dong, C.-f. Shao, C.-X. Zhuge, and M. Meng, "Spatial and temporal characteristics for congested traffic on urban expressway," *Journal of Beijing Polytechnic University*, vol. 38, no. 8, pp. 128–132, 2012.

- [3] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.
- [4] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [6] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.
- [7] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.
- [8] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.
- [9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
- [10] 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.
- [11] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.
- [12] N. L. Nihan and K. O. Holmesland, "Use of the box and jenkins time series technique in traffic forecasting," *Transportation*, vol. 9, no. 2, pp. 125–143, 1980.
- [13] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, 2013.
- [14] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [15] Y. Kamarianakis and P. Prastacos, "Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches," *Transportation Research Record*, vol. 1857, no. 1, pp. 74–84, 2003.
- [16] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.
- [17] G. Petneházi, "Recurrent neural networks for time series forecasting," *arXiv preprint arXiv:1901.00069*, 2019.
- [18] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "Dnn-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2016, pp. 1–4.
- [19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [20] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
- [21] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
- [22] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.
- [23] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.