Spatial-temporal Graph Attention Networks for Traffic Flow Forecasting

Chang Wei\textsuperscript{1,}\textsuperscript{a}, Jin Sheng\textsuperscript{2,}\textsuperscript{b}

\textsuperscript{1}Institute of Intelligent Transportation Systems Zhejiang University Hangzhou 310058, Zhejiang, China
\textsuperscript{2}Institute of Intelligent Transportation Systems Zhejiang University Hangzhou 310058, Zhejiang, China
\textsuperscript{a}Corresponding author: 21712237@zju.edu.cn, \textsuperscript{b}jinsheng@zju.edu.cn

Abstract— In order to accurately forecast the road section traffic volume, in this study, a spatial-temporal graph attention network model (GALSTM), which is based on graph attention architecture and long and short memory network (LSTM), is proposed to predict the traffic volume of road section. LSTM network is used to extract the temporal correlation of traffic flow data, and graph attention network is used to get adaptive adjacency matrix at each time step to capture the spatial correlation of road network. The proposed GALSTM model and other frequently-used traffic flow prediction methods were validated by using dataset collected by the California highway administration PeMS. Experimental results on two traffic datasets indicate that GALSTM model achieves the best prediction accuracy in all three evaluation metrics of mean absolute errors, mean absolute percentage errors, and root mean squared errors. GALSTM model can be used as an effective method to forecast the traffic volume of road section.

1. INTRODUCTION

Intelligent transportation system has become an important part of urban governance modernization and attracted more and more attention. Traffic flow prediction is one of the important contents of intelligent transportation system. Timely and accurate traffic prediction information can help us make better travel decisions, alleviate traffic congestion, reduce environmental pollution caused by traffic operation, and ensure efficient and safe road passage. Nowadays, the traffic data that can be collected is increasingly rich, and we have entered the era of big traffic data. How to mine and use the rich traffic big data, make more accurate and timely traffic flow prediction, help managers to develop better traffic control scheme, and provide support for traffic congestion management, is an important topic.

The goal of traffic flow forecasting is to predict the parameters of traffic state in the future based on the historical data (such as speed, flow) collected by the detector on the road network. However, due to the randomness of people's traffic trips and the periodic and nonlinear characteristics of traffic flows, how to capture the complex temporal and spatial correlation of traffic flows on road networks for prediction has always been a challenging subject. In recent decades, a large number of researchers have done a lot of research in this field and built a variety of traffic flow forecasting models. Previous studies show that existing models can be roughly divided into two categories: traditional statistical algorithm model and machine learning model. Most of the statistical algorithm forecasting models were developed and established years ago, such as historical average method, Kalman filtering model, mobile autoregressive model. At that time, the traffic conditions were relatively simple and the data scale was...
small. William et al. comprehensively studied the application of seasonal ARIMA model in traffic flow forecasting[1]. Dong et al. proposed a forecasting model based on Kalman filter theory incorporating temporal and spatial characteristics of traffic flow[2]. Guo et al. considered the similarity of changes in urban traffic flow, and used fuzzy logic to optimize the Kalman filter prediction model and applied it to urban traffic flow forecasting[3]. However, it is difficult for these models to deal with high-dimensional spatio-temporal sequential data. As the scale and types of traffic data that can be collected become more and more abundant, more and more researches begin to focus on models based on machine learning algorithms.

Machine learning and deep learning algorithms are widely used in various time series forecasting tasks because they can well deal with high-dimensional data to capture the nonlinear relationship, and have achieved excellent performance. Fan et al. combined the non-parametric regression model and the BP neural network to establish a traffic flow forecasting hybrid model, which can better reflect the time-varying and non-linear characteristics of traffic flow[4]. Xu et al. used a large number of training sample data to establish a classification regression tree model for short-term prediction of traffic flow, which was verified based on the data of the Portland state highway in the United States, and achieved good results[5]. Xie et al. introduced a multivariate statistical regression parameter estimation method to improve the K-nearest neighbor algorithm, which can accurately predict dynamic traffic flow information[6]. Gong et al. simultaneously considered the correlation of time series and the spatial correlation of adjacent road sections, and proposed a method for urban road travel time prediction based on gradient boosting regression tree model[7]. Deep learning algorithms show excellent performance in traffic flow prediction. Lv et al. used a deep stacked autoencoder (SAE) model to learn traffic flow characteristics and achieved excellent performance[8]. Huang et al. combined Deep Belief Network (DBN) and regression model to propose a new network architecture to predict traffic flow[9]. Recurrent neural network and its variants LSTM[10][11], GRU[12], etc., have been adopted by a lot of research in recent years due to their powerful ability in processing sequence data. Cui et al. proposed a deep stacked bidirectional and unidirectional LSTM network architecture for network-wide traffic speed prediction[13]. However, it is difficult for the above-mentioned models to simultaneously capture the temporal dynamic of traffic flow and the spatial dependence of the traffic road network from the traffic data. In order to better integrate spatial features, some research try to introduce convolutional neural networks to extract the spatial correlation of road networks. Ma et al. learnt the traffic network as images and proposed a model based on convolutional neural network to predict network-wide traffic speed with higher accuracy and faster training[14]. Traditional convolutional neural networks are more suitable for learning regular grid-type data, and cannot well extract the spatial features of complex topological structures of non-Euclidean geometry. Yu et al. learnt the traffic network as a graph, used graph convolution to capture the local spatial dependencies in the traffic network, and used one-dimensional convolution to model the temporal relationship, proposed a spatio-temporal graph convolutional network structure to predict traffic speed[15]. Li et al. used diffusion convolution for spatial dependency modeling, used GRU to construct DCGRU, and built an encoder-decoder framework to capture temporal correlation, proposed a diffusion convolution recurrent neural network structure to predict the speed on the road sensor network[16].

However, the graph convolutional network is suitable for processing static graph, and has its limitations when dealing with dynamic graph. Based on the above research, we propose a new traffic forecasting model called the graph attention LSTM network (GALSTM). We learn the traffic network as a general graph and use graph attention network [17] to construct an adaptive adjacency matrix to capture the complex spatial topological dependence of traffic network. We introduce LSTM network[10] to capture the temporal dynamics of traffic flow. The proposed model combines the graph attention operation and time sequence learning layers to model the spatio-temporal dependencies of traffic data. We evaluate our model on two real-world traffic datasets. Experiment results show that the proposed model outperforms multiple traffic forecasting baselines. The GALSTM model can also be applied to other spatio-temporal forecasting tasks.
The rest of the paper is organized as follows. METHODOLOGY introduces the details of our proposed method. In EXPERIMENTS, we evaluate the performance of the GALSTM by two real-world traffic datasets, including the comparison analysis with other similar models. We conclude the paper in CONCLUSION.

2. METHODOLOGY

2.1. Traffic Forecasting Problem
The goal of traffic forecasting refers to predicting the future traffic states given previously observed traffic flow from correlated sensors on the road network. The traffic network and topological relationship between sensor locations can be represented by a weighted directed graph $G = (v, e, A)$. $v$ is a set of nodes(vertices) $v_i \in v$, corresponding to the sensor stations; $e$ is a set of edges, representing the connectivity between stations; $A \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix, in this study, we use the function of the distance between sensor stations to measure the weight. We regard the traffic flow observed on $G$ as a graph signal $X \in \mathbb{R}^{N \times P}$, where $P$ is the number of node or edge attribute features (e.g., velocity). Let $x^{(t)}$ represent the graph signal observed at time $t$, the traffic forecasting problem can be considered to learn a function $F(\cdot)$ that maps $T'$ time steps of historical graph signals to the next $T$ subsequent time step of graph signals:

$$F \left( [X^{(t-T+1)}, \ldots, X^{(t)}]; G(v, e, A) \right) = [X^{(t+1)}, \ldots, X^{(t+T)}]$$ (1)

2.2. Spatial Dependence Modeling

![Figure 1. Graph Convolution Diagram. Assuming f(i) is a central node, the graph convolution in the GCN model is encoding the weighted sum of the first-order neighbor features on f(i) to obtain its spatial dependence.](image)

In order to capture the spatial topological relationship in the traffic road network, many forecasting models[15-16] use graph convolutional network GCN[18] to extract spatial features from traffic flow data. The GCN model constructs a filter in the Fourier domain, the filter acts on the nodes of graph and its first-order neighborhood to capture spatial features between nodes. A more intuitive view shown in Figure 1, the graph convolution in the GCN model is encoding the weighted sum of the first-order neighbor features on each vertex to obtain spatial dependence. We introduce the notion of graph convolutional operator “$\ast_G$”, as the multiplication of a graph signal $x$ with kernel $\Theta$. The graph convolution in the GCN model can be expressed as:

$$\Theta \ast_G x = \Theta \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \right) x$$ (2)

where $\Theta$ is a matrix of filter parameters, $x$ is the input feature, $A$ represents the adjacency matrix, the adjacency matrix stores the relationship between each node and its adjacent nodes in a graph, $I_n$ is an identity matrix, $\tilde{A} = A + I_n$ represents a matrix with self-connection structure, $\tilde{D}$ is the diagonal degree matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. In this way, spatio-temporal traffic network data is reorganized as graph-structured frame to extract meaningful patterns and features in the space domain.

The graph convolutional network GCN uses a static adjacency matrix, in traffic flow forecasting, the weight value is usually calculated by the following formula:

$$W_{ij} = \begin{cases} \exp\left(-\frac{\text{dist}(v_i, v_j)^2}{\sigma^2}\right), & i \neq j \text{ and } \exp\left(-\frac{\text{dist}(v_i, v_j)^2}{\sigma^2}\right) \geq \varepsilon \\ 0, & otherwise \end{cases}$$ (3)
where $W_{ij}$ represents the connection weight value of node $i$ and node $j$, $dist(v_i, v_j)$ represents the distance between the detector $v_i$ and $v_j$ on the road network, $\sigma$ is the standard deviation, $\epsilon$ is the threshold. During traffic non-rush hours, it is possible to use spatial distance to measure the weight of neighboring points. However, in peak hours, only using spatial distance to represent the weight of adjacent nodes on the road cannot fully reflect the changes of the traffic network load.

Graph convolutional networks have certain limitations when dealing with dynamic graphs. In this research, we introduce Graph Attention Networks (GAT)[17] to capture dynamic spatial dependence. Graph attention follows the self-attention strategy[19], which uses the attention mechanism to determine the weights of nodes and adjacent nodes when aggregating node feature information. As shown in Figure 2, the feature information of the nodes is used to calculate the attention coefficient of each adjacent node. The input of the GAT layer is the feature of the node, $x = \{x_1, x_2, \ldots, x_N\}, x_i \in \mathbb{R}^F$, where $N$ is the number of nodes and $F$ is the feature number of the nodes. We consider the influence of adjacent nodes at every time step. For any node $i$, its feature at time $t$ is expressed as $x_t(i) \in \mathbb{R}^F$, and the node $j \in NB(i)$ is the adjacent node of $i$. The attention coefficient $A_t[i,j]$ indicates how much attention should be allocated to the feature of node $j$ in order to predict the feature of node $i$ at time $t$, that is, the importance of node $j$ to node $i$. The graph attention coefficient can be calculated by the following formula:

$$A_t[i,j] = \frac{\exp(\text{LeakyReLU}(a^T[Wx_t(i)||Wx_t(j)]))}{\sum_{k \in NB(i)}\exp(\text{LeakyReLU}(a^T[Wx_t(i)||Wx_t(k)]))}$$

(4)

where $W$ is the transformation matrix of node features, $a(\cdot, \cdot)$ is the self-attention mechanism[19], and it is a single-layer feedforward neural network, $^T$ represents transposition and $||$ is the concatenation operation.

First, we use the mapping matrix $W$ to transform the feature of each node and its adjacent nodes, and concatenate the result of transformation, $Wx_t(i)$ and $Wx_t(j)$, then feed it into the self-attention network. We apply Leaky Rectified Linear Units (LeakyReLU) as activation function. In order to make it obvious to compare the attention coefficients of different adjacent nodes, the acquired attention coefficients are standardized by the softmax function to obtain $A_t[i,j]$.

Multi-heads attention[19] allows the model to learn attention coefficients in multiple mapping subspaces. Each attention network can capture different connections between nodes, making the attention network more robust[17,20]. The multi-head attention mechanism in GAT uses a total of $C$ attention networks, each network has its mapping matrix $W$ and self-attention network $a$, $C$ independent attention networks perform the above transformations independently, the final result is obtained by averaging the node features after $C$ conversions. The calculation expression is as follows:

$$A_t[i,j] = \frac{1}{C} \sum_{c=1}^{C} \frac{\exp(\text{LeakyReLU}(a^T[Wx_t^{(c)}(i)||Wx_t^{(c)}(j)]))}{\sum_{k \in NB(i)}\exp(\text{LeakyReLU}(a^T[Wx_t^{(c)}(i)||Wx_t^{(c)}(k)]))}$$

(5)
Figure 3. Multi-heads graph attention mechanism.

As shown in Figure 3, for each node on the graph, we calculate its attention coefficient with its all adjacent nodes. The attention coefficient can capture the changes of spatial correlation between nodes. Therefore, similar to the graph convolution operation[18], the attention coefficient can be employed to update node features. We use “⊗” to represent the adaptive graph convolution operation based on the graph attention mechanism, the calculation formula is as follows:

\[ X_t \otimes \Theta = \alpha(\Theta A_t X_t) \]  

(6)

where \( X_t \) is the input graph signal at time \( t \), \( \Theta \) is a matrix of filter parameters, \( \alpha \) is activation function and \( A_t \) represents the attention coefficient matrix.

2.3. Temporal Dynamics Modeling

The long and short-term memory network LSTM[17] is an improved recurrent neural network. It introduces storage units and gating mechanisms to solve the problem of vanishing gradient in the traditional RNN[21], and can well capture the long-term sequence data dependency. There are many applications of LSTM network in time series data mining[22]. The standard LSTM architecture is shown in figure 4.

Figure 4. LSTM cell architecture

At each time iteration, the LSTM cell has three inputs: the layer input, \( x_t \), the previous layer output state, \( h_{t-1} \), the previous cell output state, \( C_{t-1} \), and two outputs: the cell output state, \( C_t \), and the layer output, \( h_t \). There are three gates in a LSTM cell, including a forget gate \( f_t \), an input gate \( i_t \), and an output gate \( o_t \). LSTM uses two gates to control the contents of the cell state \( C_t \). The forget gate determines how much of the cell state at the previous moment \( C_{t-1} \) is retained to the current moment \( C_t \); the input gate determines how much of the current input \( x_t \) is saved to the cell state \( C_t \). And output
gate controls how much of the cell state $C_t$ is conveyed to the current LSTM cell output $h_t$. Due to the gated structure, LSTM can deal with long-term dependencies to allow useful information pass along the LSTM network. LSTM cell block presented in Figure 4, can be calculated using the following equations:

$$f_t = \sigma(W_{xf} X_t + W_{hf} h_{t-1} + b_f)$$  \hspace{1cm} (7)

$$i_t = \sigma(W_{xi} X_t + W_{hi} h_{t-1} + b_i)$$  \hspace{1cm} (8)

$$o_t = \sigma(W_{xo} X_t + W_{ho} h_{t-1} + b_o)$$  \hspace{1cm} (9)

$$\tilde{C}_t = \tanh(W_{xc} X_t + W_{hc} h_{t-1} + b_c)$$  \hspace{1cm} (10)

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$  \hspace{1cm} (11)

$$h_t = o_t * \tanh(C_t)$$  \hspace{1cm} (12)

where $f_t, i_t, o_t, \tilde{C}_t, C_t, h_t$ respectively are forget gate, input gate, output gate, instant state, cell state and output item. $X_t$ is the input at the current moment, $h_{t-1}$ is the output at the previous moment; $W$ term denote weight matrices (e.g. $W_{xf}$ maps layer input to the forget gate), $b$ term denote bias vectors (e.g. $b_f$ is the forget gate bias vector). $\sigma$ is the sigmoid activation function, tanh is the hyperbolic tangent function, and “*” represents element-wise multiplication.

2.4. Spatial Temporal Graph Attention Network

To effectively capture the spatio-temporal correlation of traffic network data, we integrate the learned attention coefficient matrix into the LSTM network, and propose the GALSTM model that can solve the problem of spatio-temporal traffic flow forecasting. Specifically, when the LSTM network combines the input $X_t$ at the current moment and the output $h_{t-1}$ at the previous moment, we replace the matrix multiplication of the fully connected operation with the adaptive graph convolution operation based on the graph attention mechanism. The graph convolution operation here does not use the original static adjacency matrix of node connection information, but an adaptive adjacency matrix updated by fusing the attention coefficient learned at each time step, which can capture the changes in the spatial correlation of nodes over time.

Figure 5. GALSTM network architecture

Figure 5 demonstrates the details of the proposed model GALSTM. The key equations are shown in (13) - (18) below:

$$f_t = \sigma(W_{xf} \otimes X_t + W_{hf} \otimes h_{t-1} + b_f)$$  \hspace{1cm} (13)

$$i_t = \sigma(W_{xi} \otimes X_t + W_{hi} \otimes h_{t-1} + b_i)$$  \hspace{1cm} (14)

$$o_t = \sigma(W_{xo} \otimes X_t + W_{ho} \otimes h_{t-1} + b_o)$$  \hspace{1cm} (15)

$$\tilde{C}_t = \tanh(W_{xc} \otimes X_t + W_{hc} \otimes h_{t-1} + b_c)$$  \hspace{1cm} (16)

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$  \hspace{1cm} (17)

$$h_t = o_t * \tanh(C_t)$$  \hspace{1cm} (18)

where “$\otimes$” represents adaptive graph convolution operation based on the graph attention mechanism.

The GALSTM model combines the sequence processing capability of the LSTM network and the spatial feature extraction capability of the GAT model. The entire network is trained by minimizing the error between the real value and the prediction using backpropagation through time. In summary, the GALSTM model is able to deal with the complex spatial dependences and temporal dynamics and can be applied to various spatio-temporal forecasting problems.
3. EXPERIMENTS

3.1. Data Description
In this study, we evaluated the performance of the models on two real-world highway traffic datasets, which were collected by the California Department of Transportation Performance Measurement System (PeMS). The dataset is collected from inductive loop detectors, and the time interval is 5 minutes, which contains three characteristics of traffic volume, average speed, and average occupancy rate. We select traffic volume data for prediction, and the geographic location information of the detectors is also included in the dataset. (1) PeMSD4 This dataset comes from the San Francisco Bay Area and contains 307 detectors. The time range is from January to February 2018. We choose the first 46 days of traffic data as the training set, 6 days of data as the validation set, and the remaining 7 days of data as the test set. (2) PEMSD8 This dataset comes from the San Bernardino area and contains 170 detectors. The time range is from July to August 2016. We choose the first 46 days of traffic data as the training set, the 6 days of data as the validation set, and the last 10 days of data as the test set.

In both of those datasets, the linear interpolation method is used to fill missing values after data cleaning, and Z-Score normalization is applied to data input. The topology of the road graph is constructed by the deployment diagram of sensor stations. Adjacency matrix is used to describe the spatial relationship between sensor and is computed based on the distances among sensor stations in the traffic network.

3.2. Experimental Setting
Baselines We compare our framework GALSTM with the following widely used time series regression baselines, including (1) Auto-Regressive Integrated Moving Average model (ARIMA), which is widely used in time series prediction; (2) Support Vector Regression model (SVR), which uses linear support vector machine for the regression task; (3) Recurrent Neural Network with fully connected LSTM hidden units (FC-LSTM); (4) Gated Recurrent Unit model (GRU).

Evaluation Metric The performance of the proposed and the compared models are evaluated by three commonly used metrics in traffic forecasting, including: Mean Absolute Errors (MAE), Mean Absolute Percentage Errors (MAPE), and Root Mean Squared Errors (RMSE). Missing values are excluded in calculating these metrics.

\[
MAE = \frac{1}{n} \sum_{t=1}^{n} |Y_t - \hat{Y}_t| 
\]

\[
MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \times 100\%
\]

\[
RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (Y_t - \hat{Y}_t)^2}
\]

where \(Y_t\) denotes the real value and \(\hat{Y}_t\) denotes the forecasting value.

All experiments are conducted on Ubuntu16.04, CPU: Intel(R) Xeon(R) E5, GPU: NVIDIA GeForce GTX1080TI, RAM 16.0G. Pytorch1.3, Python3.7. The sequence length of input data is 12, that is, the historical data of past one hour is used to predict the traffic volume after 5 minutes, 15 minutes, and 30 minutes. The number of hidden layers of the GALSTM model is 64, and the number of multi-head attention is set to 8. We divide the dataset into training set, validation set and test set. The total number of training rounds is set to 1000 Epoch. Each model trains one epoch on the training set, then verifies it on the validation set, and records the validation error. If the validation error changes for 10 consecutive epochs less than 0.00001, the training is terminated early, and the resulting model is optimal model, and we use this optimal model to test on the test set.

3.3. Experimental Results
Table I shows the contrast of GALSTM model and other approaches for 5 minutes, 15 minutes and 30 minutes on both datasets. We observe the following phenomena based on experimental results. (1) The deep learning models based on neural network including LSTM and GRU and GALSTM perform better than the time series model ARIMA and the machine learning model SVR, which proves that deep
learning method is more suitable for processing high-dimensional nonlinear data; (2) The GALSTM model has higher prediction accuracy than LSTM and GRU that only concern the temporal relationship, which proves the role of fusing spatial correlation. GALSTM model achieve the best prediction performance regarding all evaluation metrics for all forecasting horizons, which proves the effectiveness of the GALSTM model in capturing the spatio-temporal dependence.

Table I. Performance contrast of different approaches and prediction horizons

| T     | Metric | ARIMA | SVR | LSTM | GRU   | GALSTM |
|-------|--------|-------|-----|------|-------|--------|
| 5 min | MAE    | 16.34 | 15.27 | 14.81 | 14.85 | 14.44 |
|       | RMSE   | 24.56 | 23.89 | 22.43 | 22.34 | 21.95 |
|       | MAPE   | 12.37% | 11.34% | 10.35% | 10.37% | 9.62% |
| PEMS  D8 | 15 min | MAE    | 21.76 | 19.25 | 17.15 | 17.26 |
|       | RMSE   | 29.34 | 28.43 | 27.61 | 27.51 | 24.48 |
|       | MAPE   | 15.04% | 14.79% | 13.28% | 13.38% | 11.25% |
|       | 30 min | MAE    | 24.54 | 22.89 | 21.23 | 21.33 |
|       | RMSE   | 31.23 | 29.76 | 28.24 | 28.26 | 27.68 |
|       | MAPE   | 17.56% | 16.21% | 14.98% | 15.08% | 14.06% |

Table II. Performance contrast of spatial dependency modeling

| Metrics | MAE     | RMSE    | MAPE     | MAE     | RMSE    | MAPE     | MAE     | RMSE    | MAPE     |
|---------|---------|---------|----------|---------|---------|----------|---------|---------|----------|
| GALSTM  | 14.44   | 21.95   | 9.62%    | 16.22   | 24.48   | 11.25%   | 18.53   | 27.68   | 14.06%   |
| GC-LSTM | 14.68   | 22.11   | 9.96%    | 16.61   | 25.08   | 11.85%   | 18.72   | 27.74   | 14.46%   |
| Conv-LSTM | 14.75  | 21.94   | 10.07%   | 16.86   | 25.21   | 11.98%   | 18.83   | 27.86   | 14.58%   |

Table III. Performance contrast of temporal dependency modeling

| Metrics | MAE     | RMSE    | MAPE     | MAE     | RMSE    | MAPE     | MAE     | RMSE    | MAPE     |
|---------|---------|---------|----------|---------|---------|----------|---------|---------|----------|
| GALSTM  | 14.44   | 21.95   | 9.62%    | 16.22   | 24.48   | 11.25%   | 18.53   | 27.68   | 14.06%   |
| GAT     | 21.56   | 29.64   | 15.34%   | 24.78   | 35.38   | 20.45%   | 28.46   | 39.98   | 24.32%   |
3.4. Effect of Spatial Dependency Modeling

To further verify the effect of spatial dependency modeling, we compare the GALSTM model with its variants: ConvLSTM, which uses the traditional convolutional operator to capture the traffic network spatial dependence. As shown in Table II, we can clearly see that GALSTM has lower validation error, demonstrating that GALSTM model can better capture network spatial features from traffic data. Figure 6 shows the error change curve of each model at each epoch on the validation set. Due to the early termination strategy used in the training, the number of training rounds of each model is different.

3.5. Effect of Temporal Dependency Modeling

To evaluate the effect of GALSTM to extract temporal features from traffic data, we compare the proposed model with the graph attention model GAT. The original GAT model is designed for classification issues, we change its output activation function to make it suitable for regression problem. Table III shows that GALSTM achieves better forecasting precision, indicating the proposed model can capture temporal dependence well. The GAT model mainly considers the spatial features and ignores that the traffic data is typical time series data.

3.6. Model Interpretation

To better understand the GALSTM model, we select a detector on the PEMS8 dataset and visualize the 5-minutes forecasting results on the test dataset. It can be seen from Figure 7 that the GALSTM model can better capture the high and low peaks of traffic flow and the sudden change of traffic flow state, and the predicted results are stable, reflecting the good performance of the GALSTM model.

Figure 6. Validation loss of spatial dependency modeling

Figure 7. The visualization results for prediction horizons of 5 minutes of GALSTM model
4. CONCLUSION
In this paper, we learn the traffic network as a graph and propose a new model (GALSTM) for traffic forecasting, which combines the characteristics of the graph attention network and LSTM network to construct a spatio-temporal graph attention network. We use graph attention network to construct an adaptive adjacency matrix to capture the complex spatial topological dependence of traffic network and introduce LSTM to extract the dynamic temporal features of traffic time series data. Finally, the GALSTM model is used to tackle spatio-temporal traffic forecasting tasks. When evaluated on two real-world traffic datasets, our approach outperforms the baselines. The results show that the GALSTM model has the best prediction accuracy and can effectively capture sudden changes in traffic flow, which proves the excellent performance of the GALSTM model. It can be used as an effective method for traffic forecasting. For future work, we will attempt to explore the attention mechanism to extract temporal and spatial features.

REFERENCES
[1] B. M. Williams, L. A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results[J]. Journal of Transportation Engineering, 2003,129(6): 664-672.
[2] C. J. Dong, C. F. Shao, X. M. Zhou, Kalman filter algorithm for short-term jam traffic prediction based on traffic parameter correlation[J]. Journal of Southeast University, 2014,44(2):413-419.
[3] H. H. Guo, L. J. Fang, L. Yu, A short-term traffic flow prediction model based on fuzzy Kalman filtering[J]. Journal of Zhejiang University of Technology, 2013,41(2):218-221.
[4] N. Fan, X. M. Zhao, M. Dai, et al. Short-term traffic flow prediction model[J]. Journal of Traffic and Transportation Engineering, 2012,12(4):114-119.
[5] Y. Y. Xu, X. Zhai, Q. J. Kong, et al. Short-term prediction method of freeway traffic flow. Journal of Traffic and Transportation Engineering, 2013,13(2):114-119.
[6] H. H. Xie, X. H. Dai, Y. Qi, Improved K-nearest neighbor algorithm for short-term traffic flow forecasting. Journal of Traffic and Transportation Engineering, 2014,14(3):87-94.
[7] Y. Gong, X. Q. Luo, D. H. Wang, Urban travel time prediction based on gradient boosting regression trees. Journal of Zhejiang University: Engineering Science, 2018,52(3):453-460.
[8] Y. S. Lv, Y. J. Duan, W. W. Kang, Traffic flow prediction with big data: A deep learning approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2014,16(2):865-873.
[9] W. H. Huang, G. J. Song, H. K. Hong, Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2014,15(5):2191–2201.
[10] S. Hocheriter, J. Schmidhuber, Long Short-Term Memory[J]. Neural Computation, 1997,9(8):1735-80.
[11] X. L. Ma, Z. M. Tao, Y. H. Wang, Long and short-term memory neural network for traffic speed prediction using remote microwave sensor data[J]. Transportation Research Part C: Emerging Technologies, 2015,54,187-197.
[12] K. Cho, B. V. Merrienboer, C. Gulcehre, Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv Prepr, 2014, arXiv1406.1708.
[13] Z. Y. Cui, R. M. Ke, Z. Y. Pu, Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. arXiv Prepr, 2018, arXiv1801.02143.
[14] X. L. Ma, Z. Dai, Z. B. He, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction[J]. Sensors, 2017,17(4):818.
[15] B. Yu, H.T. Yin, Z. X. Zhu. Spatio-temporal graph convolutional networks:A deep learning framework for traffic forecasting. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018.
[16] Y. G. Li, R. Yu, C. Shahabi, Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In International Conference on Learning Representations, 2018.
[17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. In Proceedings of the International Conference on Learning Representations, 2017.

[18] T. N. Kipf, M. Welling. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations, 2017.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need, In Advances in Neural Information Processing Systems, 2017, pp.5998–6008.

[20] D. Busbridge, D. Sherburn, P. Cavallo. Relational graph attention networks. arXiv preprint, 2019, arXiv:1904.05811.

[21] G. Hinton, L. Deng, D. Yu. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal Processing Magazine, 2012, 29(6):82–97.

[22] D. Bahdanau, K. Cho, Y. Bengio. Neural machine translation by jointly learning to align and translate. In Proceedings of the International Conference on Learning Representations, 2015.