Spatial-Temporal Multi-Head Attention Networks for Traffic Flow Forecasting

Zhao Zhang*
School of Computer, Central China Normal University, Wuhan, Hubei, China,
925762735@qq.com

Ming Liu
School of Computer, Central China Normal University, Wuhan, Hubei, China,
lium@mail.ccnu.edu.cn

Wenquan Xu
School of Computer, Central China Normal University, Wuhan, Hubei, China,
464129748@qq.com

ABSTRACT

Traffic flow forecasting plays an important role in the intelligent traffic system, which is the basis for traffic control and traffic management. However, due to the complex spatial-temporal dependence, traffic flow forecasting has always been a difficulty in the field of intelligent traffic. In order to select a suitable spatial-temporal forecasting method and solve the problem that recurrent neural architecture is not conducive to parallel computing, we construct a spatial-temporal forecasting model by using multi-head attention models. Use graph attention networks with multi-head attention mechanism to capture spatial features, and use the scaled dot product attention with positional encoding like Transformer to capture temporal features. Experimental results on two real-world datasets demonstrate that the forecasting error of our method is lower than baseline methods.

CCS CONCEPTS

• **Computing methodologies**; • **Artificial intelligence**;

KEYWORDS

Intelligent traffic, Deep learning, Multi-head attention, Graph attention network

ACM Reference Format:
Zhao Zhang, Ming Liu, and Wenquan Xu. 2021. Spatial-Temporal Multi-Head Attention Networks for Traffic Flow Forecasting. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3487075.3487102

1 INTRODUCTION

Intelligent Traffic System (ITS) is a comprehensive transportation system which combines artificial intelligence, automatic control, sensor technology and other scientific and technological means. It is an important part of the current advanced traffic management system, which can strengthen the management of vehicles, improve the utilization rate of traffic facilities, reduce energy consumption, and provide security [1]. Traffic flow forecasting is one of important functions of the system. It can provide traffic network forecasting data for managers, which can be used as a scientific basis to judge traffic status. The specific types of traffic flow data include traffic speed, traffic flow, occupancy, etc.

Because of the complex spatial-temporal dependence of traffic data, it has always been a challenging task. Due to need to consider both spatial and temporal factors, it is different from the general time series forecasting problem. In terms of spatial dependence, each node affects the traffic data of its neighbor nodes, so the data changes of each traffic node should not be considered in isolation but need to pay attention to the interaction between nodes. The location and connectivity of each node is different, so the topological structure of the whole traffic network influences the overall traffic data change to a certain extent. In addition, in terms of temporal dependence, traffic data will change dynamically with time, mainly showing periodicity and trend. In the whole network topology, the dynamic change of traffic state of each node over time will have corresponding impact on the whole network.

Related researches have shifted from early only considering spatial dependence or temporal dependence to considering both [2]. Now most researches choose to use the combination of convolution neural network (and its variants) and recurrent neural network (and its variants), using the former to capture spatial features and the latter to capture temporal features [2][3]. Therefore, most of current researches mainly have focused on how to build a more effective spatial model and a more effective temporal model, and how to choose a more perfect way to combine them. However, some recurrent neural architecture models such as Long Short-Term Memory Network (LSTM) and Gated Recurrent Unit (GRU), which are commonly used as temporal models, have inherent defects because of their structural design [4].

In order to solve the above problems, a traffic flow forecasting method based on spatial-temporal multi-head attention networks is proposed. It uses Graph Attention Networks (GAT) with multi-head attention mechanism to capture spatial features, and uses scaled dot product attention with positional encoding like Transformer to capture temporal features. Our contributions are two-fold:

(1) Our method mainly uses multi-head attention models to construct spatial and temporal modules, which can capture the spatial-temporal dependence of traffic network data more effectively. And GAT can deal with Non-Euclidean data in traffic network well.

(2) Our works completely abandon RNN models and some their variants which are often used in traffic flow forecasting. All of

*Corresponding author.

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487102

our works depend on attention mechanism, so it can be highly parallelized in computation.
The rest of the paper is organized as follows. The second section reviews related researches. The third section introduces our method in detail. The fourth section shows relevant experimental results on real-world traffic datasets. The fifth section concludes this paper.

2 RELATED WORK

Although researches of traffic flow forecasting have been tried for decades, the whole work has entered a new stage after the rise of deep learning algorithms. Nowadays, traffic flow forecasting mainly uses data-driven methods. And models in these methods can be divided into three categories: statistical models, traditional machine learning models and deep learning models [3].

2.1 Statistical Models

The earliest traffic flow forecasting models forecast the future trend according to the statistical rules of historical data, like Historical Average Model (HA) and Autoregressive Integrated Moving Average Model (ARIMA) [1]. Stephanedes applied HA to urban traffic control system as early as 1981. The model takes the average value of all traffic data in a certain historical period as the forecasting result. It does not consider the relevant data features, so it is simple to calculate and has low time cost. But its forecasting accuracy is low, especially for extreme and unexpected situations. In 1976, Box et al. proposed the ARIMA model. This model is one of the most widely used time series models. It is composed of sequence difference, autoregressive model (AR) and moving average model (MA). In 1995, Hamed et al. used ARIMA model to forecast the traffic flow of urban trunk roads [5]. In addition, many variants are derived from ARIMA model, such as Kohonen ARIMA [6], subset ARIMA [7], SARIMA [8].

2.2 Traditional Machine Learning Models

In classification works, Support Vector Machine (SVM) has a good performance. Support Vector Regression (SVR) is based on SVM to deal with regression problems. In 2013, Jeong et al. proposed an online learnable weighted SVR method for traffic flow forecasting [9]. In 2014, Hu et al. proposed a traffic flow forecasting method based on support vector regression, and used particle swarm optimization algorithm to optimize the model parameters [10].

2.3 Deep Learning Models

In recent years, with the rapid development of deep learning, more and more researchers began to try to apply various neural network models to traffic forecasting, and achieved good results. Neural network model has attracted much attention because it can capture dynamic features of traffic data and has achieved the best effect known so far. In [11], a LSTM-based Deep Neural Network (DNN) model for traffic flow forecasting was proposed. And autocorrelation coefficient was added to improve the forecasting accuracy of the model. In [12], a traffic flow forecasting model based on Convolutional Neural Networks (CNN) was proposed. The advantages of CNN in capturing local dependence and being insensitive to data noise were used to improve the forecasting accuracy. On the issue of time series forecasting of images, the spatial-temporal forecasting model, convLSTM, was proposed in 2015 [13], which was used the combination of CNN and LSTM. At present, many traffic flow forecasting methods refer to idea of convLSTM to propose similar methods [14-16]. According to the idea of capturing the spatial and temporal dependence of the traffic network data, and considering Non-Euclidean characteristics of network data, Zhao et al. proposed the T-GCN model [2] by combining Graph Convolutional Network (GCN) and GRU, which obtained good results in the experiment. Since the traffic flow forecasting work is similar to the machine translation, researchers begin to build the encoder-decoder model [17] based on RNN-like units, and then add various CNN-like networks to capture spatial information. At present, many researchers have adopted such methods [3][18]. In addition, due to the success of attention model in the field of computer vision and natural language processing, more and more researchers have begun to apply it to traffic flow forecasting works [3] [19] [20].
The method based on statistical model is simple and easy to calculate. However, these models can not reflect the nonlinearity and uncertainty of traffic data well, so they are easily affected by unexpected traffic accidents and their calculation accuracy is also low. Most of methods based on traditional machine learning focus on learning only spatial dependence or only temporal dependence, instead of considering both two problems. In addition, the forecasting accuracy also needs to be improved. Nowadays, more and more researchers begin to consider combining a variety of neural networks to establish a model that has a good ability of learning spatial dependence and temporal dependence. Most researchers tend to choose a pattern of CNN-like + RNN-like. On this basis, many researchers also add various attention mechanisms to improve the accuracy of the algorithm.
Considering that computing method based on RNN-like units has the obstacle of parallelization, our works depart from recurrent neural architecture models of many studies. This paper proposes a traffic flow forecasting method that uses multi-head attention mechanism to capture the spatial-temporal dependence. Our object is to improve forecasting accuracy and parallelization degree.

3 METHODOLOGY

Our model still focuses on how to capture spatial and temporal dependences. The paper first gives an abstract definition of the problem, and then explain our temporal attention module and spatial attention module in detail. Finally, introduce the overall architecture of proposed model.

3.1 Problem Definition

First of all, make an abstract definition to make the problem generic. G is the graph of road network. The graph, $G = (V, E)$, represents the whole network topology, where V is the set of vertices, E is the set of edges. In different problems, the representation of traffic network is different. There are some studies that take nodes as V and roads as E, and others that take roads as V and nodes as E. A is the adjacency matrix of the network. A_{ij} is the weight between vertex V_i and vertex V_j, which can be calculated in many ways. X is the data of the whole network over a period of time. $X^{N\times P}$ represents the traffic state set of N vertices in P time steps. And X_i

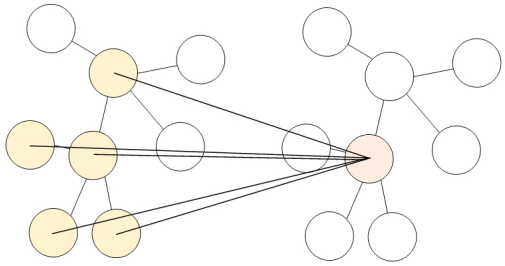

Figure 1: Convolution Operation of GAT.

represents the data vector of all N nodes in time step i. X^{i} represents the data vector of all P time steps in node i. The status value may be traffic density, traffic speed, traffic flow, etc. Our work is using the whole traffic network data of N time steps to forecast the data of the next M time steps. The formula is

$$[X_{t+1}, \ldots, X_{t+m}] = f\left(G, [X_{t-n+1}, \ldots, X_t]\right) \tag{1}$$

3.2 Spatial Attention Module

The traditional CNN is able to extract features from a regular two-dimensional matrix such as images and achieve a good result. However, the complex network topology is a graph structure, which cannot work well by the same method. Because the former is Euclidean structure, and the latter is Non-Euclidean structure, which means that the features of network and image are quite different. Because the traditional CNN cannot deal with Non-Euclidean data well, most researchers choose to use graph convolution network. The core of GCN is completing spectral decomposition by Laplacian matrix [21]. Its each convolution directly updates the parameters of the whole graph. Laplacian matrix is calculated from adjacency matrix and the weight of adjacency matrix is usually calculated by the distance between nodes. Because the shape, curvature, texture and so on of roads between nodes may be different, using distance cannot reflect the correlation between nodes well. This paper uses GAT [22] to construct spatial attention modules.

GAT uses attention mechanism to calculate the weight between two nodes. The formula is expressed as

$$e_{ij} = \alpha\left(WX^{\mathrm{i}}, WX^{j}\right) \tag{2}$$

$$\alpha_{ij} = \mathrm{softmax}_j\left(e_{ij}\right) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \tag{3}$$

Where $\alpha()$ is

$$\alpha\left(WX^{\mathrm{i}}, WX^{j}\right) = LeakyReLU\left(a^T\left[WX^{\mathrm{i}} || WX^{j}\right]\right) \tag{4}$$

$\mathcal{N}_i$ represents the collection of node i and its one-hop neighbor nodes. a is a single-layer feedforward layer. After calculating the weight, each node is convoluted on its one-hop neighbor nodes and itself. The process is shown in Figure 1. Multi-head attention mechanism can improve feature extraction ability of GAT [22], and the formula is

$$X^{j\prime} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{k} W^{k} X^{j}\right) \tag{5}$$

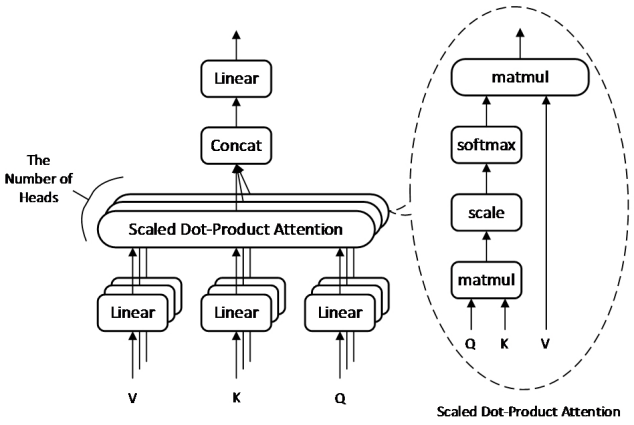

Figure 2: Temporal Multi-Head Attention [4].

3.3 Temporal Attention Module

LSTM [23] and GRU [17] are commonly used in solving problems of time series forecasting. In 2014, K Cho et al. proposed seq2seq model [17]. In the same year, [24] proposed the attention model for seq2seq. Since then, the model has been widely used in the field of natural language processing (NLP). Subsequently, more and more researchers begin to apply the model to the field of traffic flow forecasting [3][18]. However, RNN-like models and seq2seq based on RNN-like units have difficulties in parallelization. In 2017, Google proposed the Transformer model [4], which improved accuracy in many tasks of NLP. In addition, Transformer abandons RNN-like units and only uses encoder-decoder model and attention mechanism, which makes the whole model parallelized efficiently. This paper refers to Transformer and uses multi-head attention model to extract temporal features. Divide the input data into three parts: Q (Query), K (Key) and V (Value). Different from Transformer, this paper does not use encoder-decoder structure so this paper chooses to remove mask. The calculation process is shown in Figure 2. The scaled dot product attention is shown in formula (6), where d_k is the dimension of Key.

$$Attention\left(Q, K, V\right) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

Different from recurrent neural architecture, attention is not sensitive to location, so it needs to add locational encoding as in Transformer. Our method uses sin() and cos() to add position information to the temporal module, as shown in formula (7) and formula (8). Where pos is the serial number of the time series entered in the batch. d_{model} represents the dimension of the data in a certain time step. If dimension of the input vector is not reduced, it is also equal to the number of traffic nodes. The (2i)th dimension location information corresponding formula (7) and the (2i+1)th dimension location information corresponding formula (8). Finally, add the position information matrix to the input matrix of positional encoding.

$$PE\left(pos, 2i\right) = sin(pos/10000^{2i/d_{model}}) \tag{7}$$

$$PE\left(pos, 2i+1\right) = cos(pos/10000^{2i/d_{model}}) \tag{8}$$

3.4 Spatial-Temporal Model

The spatial-temporal model is shown in Figure 3. Our model consists of N same main modules and an FC layer, in which the main module is used to capture the features in the data and linear is used to transform the final size to be consistent with the output. The main module is divided into three parts: spatial attention module, temporal attention module and feed forward module. Among the three parts, we add Add&Norm layer, which is composed of residual connection and normalization. Its main function is to speed up the training speed and solve the problem of gradient disappearance. Finally, dropout is added to each module to prevent over-fitting.
The structure of our GAT module is shown in Figure 4. The GAT module has two layer GATs, where the first layer is a multi-head attention GAT and the second layer is a single head attention GAT. In this way, combining the linear of the output part of the first layer and the linear of the input part of the second layer into one. Output part of GAT module is elu activation function.

$$F(X) = \max(0, XW_1 + b_1)W_2 + b_2 \tag{9}$$

After building the spatial attention module and the temporary attention module, add the feed forward module, which is mainly used for dimension transformation. Our feed forward network consists of a linear layer, a relu function and a linear layer at the end, as shown in formula (9).

4 EXPERIMENTS

4.1 Dataset

In the selection of datasets, we use the dataset in [25], PeMSD7(M), to forecast the traffic speed and the dataset in [19], PeMSD4, to forecast the traffic flow.
PeMSD7(M). Caltrans performance measurement system (PeMS) collects real-time data in the major urban parts of the California highway system. The data is aggregated into 5-minute interval from 30-second data samples. And PeMSD7(M) selects an area in District 7 of California as the data source, with 228 monitoring stations. The weight of adjacency matrix is calculated by distance.
PeMSD4. The dataset is the data in District 4 obtained by Caltrans PeMS. It is obtained from 3848 detectors on 29 roads in the San Francisco Bay area. But [19] chose to remove some redundant sensors that were too close to each other, and finally reduced the number of sensor nodes to 307. All traffic data were aggregated every 5 minutes from January to February in 2018.

4.2 Experimental Settings

Our all experiments are conducted on a Linux server (CPU: Intel(R) Xeon(R) CPU E5-2643 v4 @ 3.40GHz×24, GPU: NVIDIA TITAN Xp/PCIe/SSE2). Our experiment is done by Pytorch. Train set : validation set : test set is equal to 6:2:2 on our dataset. Experiments use the historical data of 20 time steps to forecast the future data of 8 time steps. The loss function of training process is MSELoss and optimizer is Adam. The initial learning rate is set to 0.001. Batch size is set to 32. Weight decay is 1e-5. In LearningRateScheduler, step_size is 50 and gamma is 0.1. Set epoch to 240, and each baseline algorithm can converge in this number of iterations.

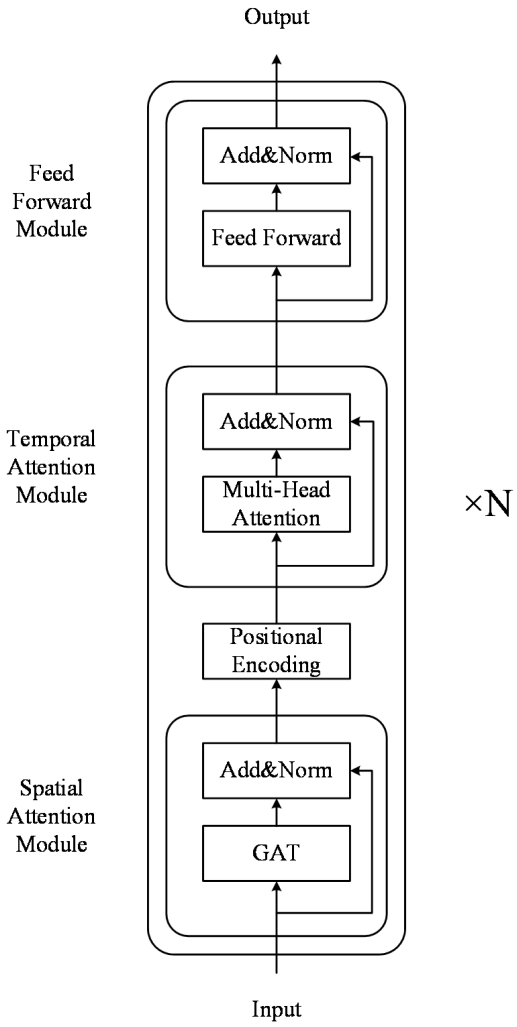

Figure 3: Spatial-Temporal Multi-Head Attention Networks.

4.3 Evaluation Metrics and Baselines

In the evaluation, use three widely used indicators: mean square error (MSE), root mean square error (RMSE) and mean absolute error (MAE).
Among the methods based on statistics and traditional machine learning, choose HA, ARIMA and SVR as baseline algorithms. In the deep learning method, because different researchers use different models and model combination methods, and each research will add its own unique way of data processing, it is difficult to fully reproduce. Therefore, here we will summarize many studies and establish a more unified or common model architecture: DL-Baseline1 and DL-Baseline2.
DL-Baseline1: The baseline refers to [2][3]. Like T-GCN model, choose GCN and GRU to build the baseline. But DL-Baseline1 is a

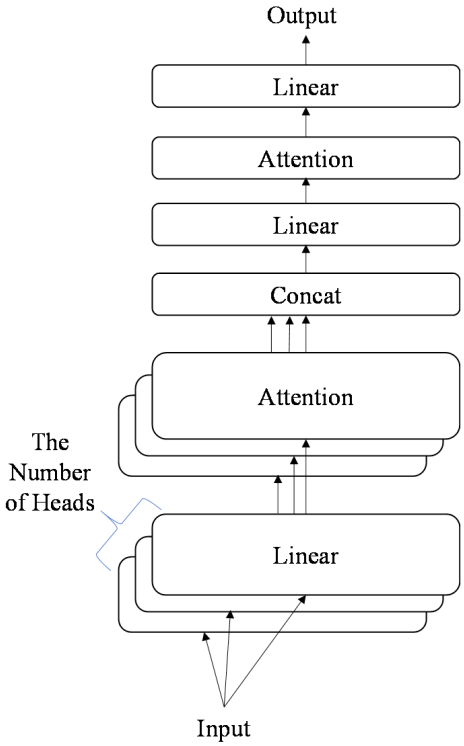

Figure 4: GAT Module.

two-layer GCNs + seq2seq model, in which the seq2seq part has the attention mechanism and is constructed by a single-layer and unidirectional GRU. We choose to use the last time step data of the encoder_input as the initial input of the decoder.

DL-Baseline2: We refer to [20] [26] to build a model of two-layer GATs + two-layer LSTMs + one-layer FC. Set the number of heads of GAT to 6 and 1 on two datasets respectively, and add Add&Norm for GAT module.

4.4 Experimental Results

Because the value of original adj on datasets is the distance between nodes. Therefore, it is necessary to preprocess it to facilitate the subsequent calculation of GCN and GAT, and the data matrix also needs to be normalized. Use formula (10) to process the original data of the data matrix and adjacency matrix on datasets. Where v_{min} is the smallest value in the matrix and v_{max} is the largest value in the matrix. Then we need to process the adjacency matrix, as shown in formula (11). d'_{ij} is the result of d_{ij} calculated by formula (10) and d_{ij} represents the distance between node i and node j. The parameters k and ϵ are set according to different datasets. So set the parameters of PeMSD7 (M) to 5 and 0.4 respectively according to our experiments. On PeMSD4 dataset, because the adjacency matrix is sparse, we set k and ϵ to 5 and 0 respectively.

$$v' = \frac{v - v_{min}}{v_{max} - v_{min}} \tag{10}$$

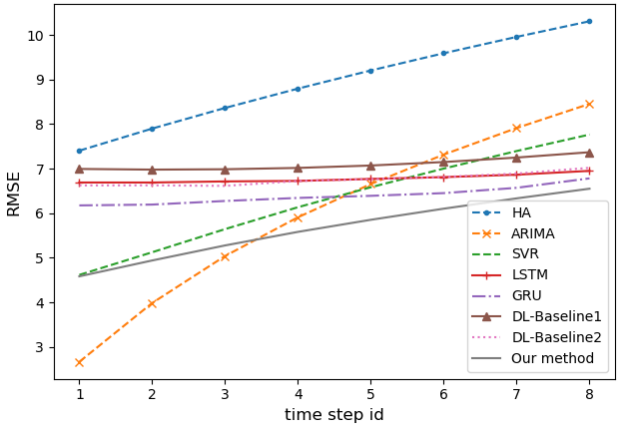

Figure 5: Comparison of Forecasting Results of Different Time Step ID on PeMSD7(M) Dataset.

$$adj_{ij} = \begin{cases} \exp\left(-kd'_{ij}\right), i \neq j \text{ and } \exp\left(-kd'_{ij}\right) \geq \epsilon \\ 0, otherwise \end{cases} \tag{11}$$

On PeMSD7(M), Set the number of heads in our spatial module and temporal module to 6 and 3 respectively. On PeMSD4, only use single head attention. Both the main module and FC are one layer. In the experiment, the MSE, RMSE and MAE of each algorithm are shown in Table 1 and Table 2. Where LSTM model is two layers LSTMs with a layer FC and GRU model is two layers GRUs with a layer FC. From Table 1 and Table 2, we can observe that our model performs better than the baseline models.

Next, Figure 5 and Figure 6 show forecasting results of some better algorithms for different time step ids. From these, we can find that our method, SVR and HA are suitable for short time step forecasting and others are suitable for long time step forecasting. But in 8 time steps, the RMSE of our method is generally lower than that of other methods.

In Table 3, showing the effect of the number of heads on our model on PeMSD7(M) dataset. Although it does not mean that the more the number of heads, the higher the forecasting accuracy, the multi-head attention is still effective for improving the forecasting accuracy.

5 CONCLUSION

This research proposes a kind of spatial-temporal multi-head attention networks for traffic forecasting, which combines GAT model and Transformer model. Use graph attention networks with multi-head attention to capture spatial dependence, and use the scaled dot product attention architecture with positional encoding like Transformer to capture temporal dependence. Our model can better extract the spatial and temporal features of traffic data and has a high degree of parallelism. From the experimental results on two real-world datasets, our method is superior to others baseline methods.

Table 1: Forecasting Results on PeMSD7(M) Dataset

Method	MSE	RMSE	MAE
HA	80.85268	8.99181	4.88492
SVR	40.55771	6.36849	4.84341
ARIMA	39.40654	6.27746	3.24503
LSTM	45.93729	6.77771	3.97502
GRU	40.97248	6.40097	3.82296
DL-Baseline1	50.46270	7.10371	4.20808
DL-Baseline2	45.71355	6.76118	4.05911
Our Method	32.36701	5.68920	3.39156

Table 2: Forecasting Results on PeMSD4 Dataset

Method	MSE	RMSE	MAE
HA	3679.13346	60.65586	41.89770
SVR	2932.50904	54.15265	44.56643
ARIMA	6311.31086	79.44376	38.86168
LSTM	1551.30390	39.38660	24.48639
GRU	1566.62198	39.58058	24.72010
DL-Baseline1	1642.60854	40.52911	25.70785
DL-Baseline2	1558.35642	39.47602	25.12811
Our Method	1317.03657	36.29100	24.19169

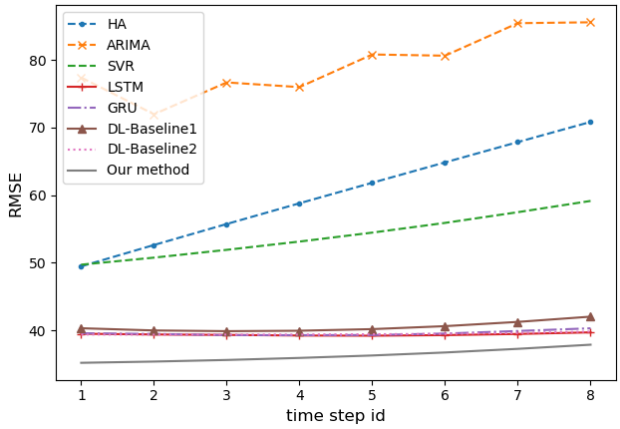

Figure 6: Comparison of Forecasting Results of Different Time Step ID on PeMSD4 Dataset.

Table 3: Forecasting Results of Different N_Heads on PeMSD7(M) Dataset

N_Heads (Spatial Module , Temporal Module)	MSE	RMSE	MAE
1,1	61.71700	7.85602	4.84636
6,1	33.90623	5.82291	3.51857
1,3	32.81613	5.72854	3.38155
6,3	32.36701	5.68920	3.39156

REFERENCES

[1] J Liu, W Guan (2004). A summary of Traffic Flow Forecasting Methods. J. Highway Transp. Res. Develop., 21(3), 82–85.

[2] L Zhao, Y Song, C Zhang, *et al.* (2019). T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. IEEE Transactions on Intelligent Transportation Systems, 21(9), 3848-3858.

[3] R D Medrano, J L Aznarte (2020). A Spatio-Temporal Attention-Based Spot-Forecasting Framework for Urban Traffic Prediction[J]. Applied Soft Computing Journal, 96.

[4] A Vaswani, N Shazeer, *et al.* (2017). Attention Is All You Need. In NIPS, 5998–6008.

[5] M M Hamed, H R Al-Masaeid, Z M B Said (1995). Short-Term Prediction of Traffic Volume in Urban Arterials. Journal of Transportation Engineering, 121(3), 249-254.

[6] M V D Voort, M Dougherty, S Watson (1996). Combining Kohonen Maps with Arima Time Series Models to Forecast Traffic Flow. Transportation Research Part C Emerging Technologies, 4(5), 307-318.

[7] S Lee, D Fambro, S Lee *et al.* (1999). Application of Subset Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting. Transportation Research Record Journal of the Transportation Research Board, 1678(1), 179-188.

[8] B M Williams, L A Hoel (2003). Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. Journal of Transportation Engineering, 129(6), 664-672.

[9] Y S Jeong, Y J Byon, M M Castro-Neto, *et al.* (2013). Supervised Weighting-Online Learning Algorithm for Short-Term Traffic Flow Prediction. IEEE Transactions on Intelligent Transportation Systems, 14(4), 1700-1707.

[10] J Hu, P Gao, Y Yao, X Xie, (2014). Traffic Flow Forecasting with Particle Swarm Optimization and Support Vector Regression, 17th International IEEE Conference on Intelligent Transportation Systems, 2267-2268.

[11] Q Zhuo, Q Li, H Yan, Y Qi, (2017). Long Short-Term Memory Neural Network for Network Traffic Prediction. International Conference on Intelligent Systems and Knowledge Engineering, 1-6.

[12] C Song, H Lee, C Kang, W Lee, Y B Kim, S W Cha, (2017). Traffic Speed Prediction under Weekday Using Convolutional Neural Networks Concepts. IEEE Intelligent Vehicles Symposium (IV), 1293-1298.

[13] X Shi, Z Chen, H Wang, *et al.* (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, MIT Press.

[14] J Bao, H Yu, J Wu, (2019). Short-Term FFBS Demand Prediction with Multi-Source Data in A Hybrid Deep Learning Framework. Intelligent Transport Systems, IET.

[15] H Yu, Z Wu, S Wang, *et al.* (2017). Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks. Sensors.

[16] X Lv, Z L Wang, Y Ren, *et al.* (2019). Traffic Network Resilience Analysis Based On The GCN-RNN Prediction Model. 2019 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering.

[17] K Cho, B V Merrienboer, C Gulcehre, *et al.* (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Computer ence.

[18] D Chai, L Wang, Q Yang, (2018). Bike Flow Prediction with Multi-Graph Convolutional Networks. 397-400.

[19] S Guo, Y Lin, N Feng, *et al.* (2019). Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 33, 922-929.

[20] T Wu, F Chen, Y Wan (2018). Graph Attention LSTM Network: A New Model for Traffic Flow Forecasting. International Conference on Information Science and Control Engineering, 241-245.

[21] J Bruna, W Zaremba, A Szlam, Y Lecun, (2014). Spectral Networks and Locally Connected Networks on Graphs. Computer Ence.

[22] P Velikovi, G Cucurull, A Casanova, *et al.* (2017). Graph Attention Networks.

[23] S Hochreiter, J Schmidhuber. (1997). Long Short-Term Memory. Neural Computation.

[24] D Bahdanau, K Cho, Y Bengio. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. Computer Science.

[25] B Yu, H Yin, Z Zhu. (2018) Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. IJCAI 2018.

[26] H Yu, Z Wu, S Wang, *et al.* (2017). Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks. Sensors.