TRAFFIC FLOW PREDICTION USING SPATIAL-TEMPORAL GRAPH NEURAL NETWORKS

Chuan Liu, Qingyuan Hou, Ruibing Song

University of Rochester

ABSTRACT

Accurate and timely traffic flow information is in high demand due to its ability to enable road users to make informed travel decisions, mitigating traffic congestion, and improving traffic management efficiency. Traffic flow prediction aims to provide this vital information, which typically lies in historical and real-time traffic data collected from various sensors. In previous studies, various models such as KNN and LSTM have been used for traffic flow prediction. However, they only focus on either spatial information or temporal information and cannot make fully use of the data. To improve the prediciton, we suggest to use Spatial-Temporal Graph Neural Networks (STGNN) and the reason is two-fold. First, the traffic itself is originally a graph structure, so we can use GNNs to make better use of the spatial information. Second, STGNNs can combine both spatial and temporal information to predict with higher accuracy. In this work, we build two STGNN models based on *PyTorch*, then train and evaluate them on two real-world traffic flow datasets. The results show that both model perform well, while MTGNN performs a little better on both datasets than GWN, resulting in 23.2-27.13 RMSE on traffic flow (per 5 minutes).

Index Terms— Traffic flow, Spatial-Temporal prediction, Graph Neural Networks

1. INTRODUCTION

Traffic congestion is a world-wide problem, which causes tremendous loss for individuals and governments. A study [1] shows that about 149 hours are wasted per vehicle due to traffic congestion, which brings about 2205 dollars to the driver. Another study [2] shows that traffic congestion brings about 305 billions dollars of economic loss to the United States government including the cost of congestion per driver as well as the extra budget on handling the extra emitted carbon.

To solve these traffic problems, accurate and timely traffic flow information is in high demand among individual travelers, businesses, and government agencies[3]. It enables road users to make informed travel decisions, mitigating traffic congestion, and improving traffic management efficiency. Traffic flow prediction aims to provide this vital information, gaining increasing attention with the rapid evolution and adoption of intelligent transportation systems. For instance, dynamic signal timing [4] technology adjusts the traffic signal timing based on predicted traffic information, making the road network work in higher efficiency. As introduced in [5], it is acknowledged as a crucial element for the successful implementation of various ITS subsystems, particularly advanced traveler information systems, advanced traffic management systems, advanced public transportation systems, and commercial vehicle operations.

The basis of traffic flow prediction typically lies in historical and real-time traffic data collected from various sensors. With the rise of intelligent traffic management and advancements in sensor technology, there has been an explosive growth in traffic data, making data-driven prediction more feasible. In previous studies, various models such as KNN and LSTM have been used for traffic flow prediction. However, we believe these models are not the best choices for this application. The reason is that the traffic flow prediction contains information in two channels: spatial information and temporal information. To be specific, the spatial information is like whether two crossroads are connected by a road or how far are they from each other. The temporal information is the previous traffic flow of each crossroads. The model KNN and LSTM, both emphasize only one of the channels. KNN is better on processing spatial information, and LSTM focus on temporal prediction. To better solve this problem, the model we used should be able to combine these two channels of information.

To better make use of both the temporal and spatial information, we suggest to use Spatial-Temporal Graph Neural Networks (STGNN) and the reason is two-fold. First, the traffic itself is originally a graph structure, i.e., the crossroads are the nodes in graph, and the roads are the edges between nodes. Therefore, we can use GNNs to make better use of the spatial information. Second, among different kinds of GNNs, STGNNs are specially designed for this kind of spatial-temporal prediction problems, combining both spatial and temporal information to predict with higher accuracy. So, we believe we can use STGNN models to predict the traffic flow with better performance than KNN and LSTM models.

STGNN is a special type of graph neural network characterized by its integration of GNNs with various temporal learning techniques to capture dynamic features across both spatial and temporal dimensions[6]. It is commonly used for data with Spatio-Temporal attributes such as transportation, environment, public safety, health, energy, economy, and other fields. There are multiple variants of STGNN in previous researches and we specifically select GWN[7] and MTGNN[8] considering there good performance and widespread recognition witnin the community. Both model are built based on *PyTorch* and trained on two real-world traffic flow datasets from the Department of Transportation of California. After hyper-parameter searching, we finally test the models on both datasets, showing good performance on Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The results show that MTGNN performs better on both datasets than GWN, resulting in 23.2-27.13 RMSE on traffic flow (per 5 minutes).

2. METHOD

In this section, we first introduce the adopted STGNN models, then clarify the training and evaluation method. As for the dataset, a data visualization is shown to better explain the features in this application.

2.1. GWN and MTGNN

As shown in Fig. 1, GWN consists of K spatial-temporal layers on the left and an output layer on the right. Input features are first transformed by a linear layer, then passed through a gated temporal convolution module (gated TCN), followed by a graph convolution network (GCN, which includes two graph convolution layers). Each spatial-temporal layer has residual connections and can be skip-connected to the output layer. The key idea of GWN is, by stacking multiple spatial-temporal layers, GWN can address spatial dependencies across various temporal scales. For instance, at the lower layers, GCN deals with short-term temporal information, while at the higher layers, GCN addresses long-term temporal information.

The framework of MTGNN is shown in Fig. 2, where a graph learning layer, m graph convolution modules, m temporal convolution modules, and an output module are connected to build the whole framework. To better learn the hidden associations among nodes, the graph learning layer computes a graph adjacency matrix, which is then used as input to all graph convolution modules. Graph convolution modules and temporal convolution modules are interleaved, capturing spatial and temporal dependencies respectively. Like GWN, MTGNN also includes Residual connections and skip connections to avoid gradient vanishing problem.

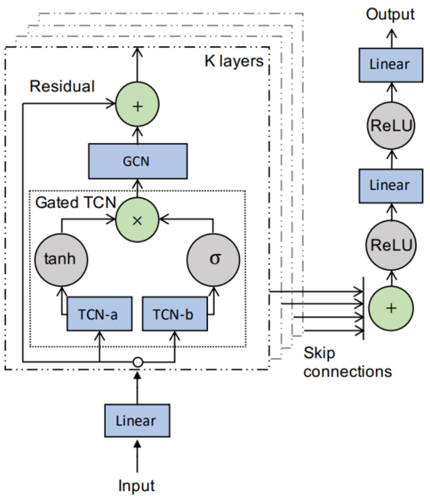

Fig. 1. The framework of GWN[7].

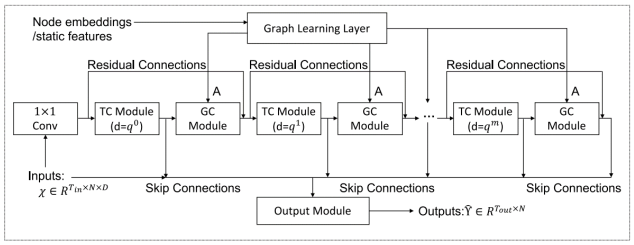

Fig. 2. The framework of MTGNN[8].

2.2. Training and Evaluation Methods

The training of the models are also implemented based on *PyTorch* library. During training, three input features are learned, including the Graph topology, the previous traffic flow, and the time embedding. The graph topology is represented by an adjacent matrix where the connections between nodes are defined according to the physical distance between the sensors. This provides the model a spatial information to conduct the graph convolution modules. The previous traffic flow contains both temporal and spatial information. The time embedding works like labels to help the model understand the temporal information.

During both training and evaluation, we use 1 time slice to predict 1 time slice, which means in the forwarding process, the input of model is only one slice of previous traffic flow and its corresponding time embedding, and the output is thus the predicted traffic flow in the next time slice.

To evaluation the performance of the models, the MAE and RMSE are calculated to measure the distance between predicted values and the ground truth. The difference is that RMSE emphasizes larger errors more than smaller ones.

2.3. Data pre-processing and Visualization

In this work, we adopt two real-world traffic flow datasets, PEMS04 and PEMS08, which come from the website of the Department of Transportation of California (https://pems.dot.ca.gov). PEMS04 records two months of traffic flow on 307 sensors on the California freeway. PEMS08 contains two months of traffic flow on 170 sensors on the California freeway. The key parameters of both datasets are listed in Table 1.

Dataset	Nodes	Time steps	Data range	Interval
PEMS04	307	16992	0-919	5min
PEMS08	170	17856	0-1147	5min

Table 1. PEMS04 and PEMS08 datasets.

To clearly show the features in the datasets, we visualize three nodes in each dataset to show the variation of traffic flow versus time. As shown in Fig. 3, the traffic flow of different sensor shows very similar temporal variation tendency.

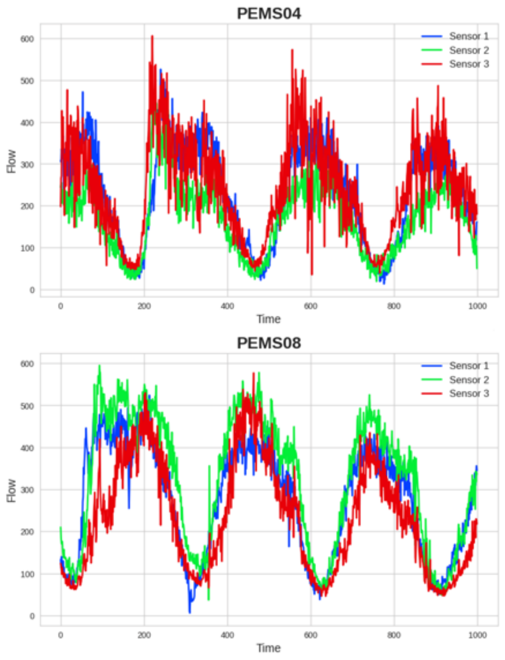

Fig. 3. Visualization of datasets.

Considering the feature of traffic flow application, two time embedding are added in our experiments, including time-in-day (1 day) and day-in-week (7 days). Both datasets are normalized with following formulation:

$$Data_{norm} = (Data - mean)/std \tag{1}$$

3. EXPERIMENTS

In this section, we first show the training loss curve of the selected models on each dataset, then report the inference accuracy on test-set. Both two datasets are split in chronological order with 70% for training, 10% for validation, and 20% for testing. The device we used is an NVIDIA RTX 3090 GPU via PyTorch GPU computing support. Recall that during both training and evaluation, we use 1 time slice to predict 1 time slice, i.e., the input of model is one slice of previous traffic flow and its corresponding time embedding, and the output is the predicted traffic flow in the next time slice.

3.1. Training loss of STGNNs

As shown in Fig. 4, on PEMS04 dataset, the training loss of both GWN and MTGNN reduces rapidly in the first epochs and then converge to a stable value. Since the lower loss is better, the curve shows that MTGNN need longer time to converge but finally get better results than GWN. This is because the network of MTGNN is more complex and have stronger presentation ability than GWN.

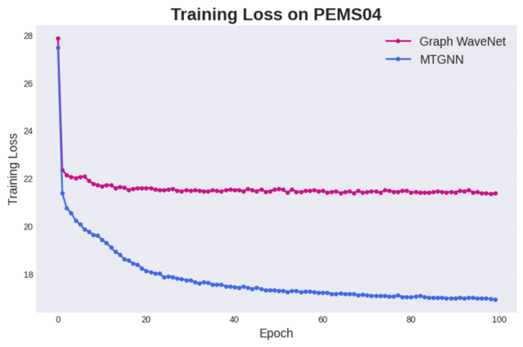

Fig. 4. Training loss of GWN and MTGNN versus epoch on dataset PEMS04.

Similar results can be achieved on PEMS08 dataset, which is shown in Fig. 5. The difference between the two dataset is, on PEMS08 both MTGNN and GWN converge very fast. This is because the nodes of PEMS08 (170) is much less than PEMS04 (307), therefore the difficulty of learning the feature is also less. In this case, MTGNN converges as fast as GWN and still have better performance.

3.2. Inference Performance

After showing the Training loss curve, we summarise the inference performance in Table 2. The results show consistency with the training loss curve, i.e., MTGNN overperforms GWN on both datasets on MAE and RMSE. Additionally, since PEMS08 has less nodes and thus easier to learn, both MTGNN and GWN get better accuracy on PEMS08 than PEMS04, showing high relationship between the prediction accuracy and traffic graph scale.

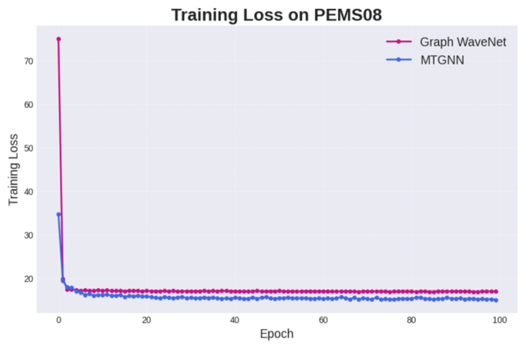

Fig. 5. Training loss of GWN and MTGNN versus epoch on dataset PEMS08.

Method	PEMS04	PEMS08
GWN-MAE	21.3659	16.9493
MTGNN-MAE	16.9265	15.0600
GWN-RMSE	33.9601	26.1909
MTGNN-RMSE	27.1356	23.2096

Table 2. Inference Accuracy of GWN and MTGNN on PEMS04 and PEMS08 datasets.

4. CONCLUSION

In this work, we implement traffic flow prediction via STGNNs which have high potential to make use of both the Spatial information and Temporal information. Two selected model, GWN and MTGNN are built and trained on two real-world traffic flow datasets PEMS04 and PEMS08. Both training loss curves and inference accuracy results show that GWN and MTGNN work well on Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). While MTGNN performs better on both datasets than GWN, it achieves MAE 21.4 (per 5 mins) on PEMS04 and 16.9 on PEMS08, showing good performance on traffic flow prediction.

5. CONTRIBUTIONS OF EACH STUDENTS

Chuan Liu: Discussion and model selection, implementation of GWN, model evaluation, mid-term presentation, report writing.

Qingyuan Hou: Discussion and model selection, dataset preprocessing, hyper-parameter searching, presentation preparation.

Ruibing Song: Discussion and model selection, implementation of MTGNN, data visualization, report writing.

6. REFERENCES

[1] Tim Levin, "The 31 us cities that had the worst traffic in 2019 according to a study," 2019.

[2] Zamira Rahim, "Here's how much sitting in traffic is costing you," Money.com, 2018, Archived from the original on August 24, 2020. Retrieved March 28, 2018.

[3] Nan Zhang, Fei-Yue Wang, Fenghua Zhu, Dongbin Zhao, and Shuming Tang, "Dynacas: Computational experiments and decision support for its," *IEEE Intelligent Systems*, vol. 23, no. 6, pp. 19–23, 2008.

[4] S. Hossan and N. Nower, "Fog-based dynamic traffic light control system for improving public transport," *Public Transport*, vol. 12, pp. 431–454, 2020, [Online; accessed Date-of-Access].

[5] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[6] Senzhang Wang, Jiannong Cao, and Philip S. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3681–3700, 2022.

[7] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2019, IJCAI'19, p. 1907–1913, AAAI Press.

[8] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, 2020, KDD '20, p. 753–763, Association for Computing Machinery.