A Key Path-Based Deep Learning Approach for Urban Traffic Speed Prediction

Wentian Chen¹,a, Jie Fang²,b,*, Zhijia Liu³, Mengyun Xu⁴

¹College of Civil Engineering, Fuzhou University, Fuzhou, People's Republic of China
²College of Civil Engineering, Fuzhou University, Fuzhou, People's Republic of China
³College of Civil Engineering, Fuzhou University, Fuzhou, People's Republic of China
⁴Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan, People's Republic of China

aN190520010@fzu.edu.cn,  b*fangjie@fzu.edu.cn

Abstract—Traffic speed prediction in the urban road network is an important and promising task of intelligent transportation systems. Precise traffic speed prediction can mitigate traffic congestion and improve road network utilization. This task is challenging because of the complexity of the spatiotemporal dependency of traffic data among road network. Existing approaches mainly focus on the whole road network and it may capture much redundant information and lead to high computational cost. In this paper, we propose a Key Path-Based Deep Learning Approach: Path-Based CNN-1D + GRU + CNN-2D (P-CGC), a novel deep learning model for traffic speed prediction. Specifically, we use EST-matching algorithm to match the float car data into the road network. Then we select several key paths and build the model for the loop detectors which are in the same key path. We introduce CNN-1D, GRU to extract the temporal dependency of the data, where CNN-1D is used to fuse the contextual information, and GRU is used to capture the features of the temporal dimension. Then we concatenate the output of all CNN-1D+GRU models and use CNN-2D to capture the spatial dependency of the data. Finally, a fully connected neural network is used to transform features into the prediction. We conduct extensive experiments on Zhangzhou real-world datasets, and the proposed approach achieves a good result.

1. INTRODUCTION

Recently, many countries try their best to develop ITS to solve the problem of traffic management. Traffic prediction is an indispensable part of ITS, especially on urban roads where traffic conditions change greatly. If accurate prediction can be made in advance, vehicles can be guided more reasonably, and the efficiency of the urban network can be improved accordingly.

Traffic speed prediction of the urban road network is a challenging spatiotemporal prediction problem. On the one hand, there are complex spatiotemporal dependencies between different links. For example, the speed of a link is often affected by its adjacent links in space, or by the links that are not adjacent but belong to the same frequently used path. On the other hand, there is a nonlinear correlation between the observations at different times. Predictions for a particular time usually require historical observations.

For the spatiotemporal prediction problem, the earliest researchers used model-driven approaches to conduct traffic prediction [1][2][3][4]. Later, autoregressive integrated moving average (ARIMA) [5][6]
and Kalman filtering [7][8][9] are widely used to realize traffic prediction tasks. Machine learning methods such as KNN [10] and SVM [11] can model more complex data. All the methods above have some limitations, and it is difficult to capture the complex spatiotemporal dependency. Because of the breakthrough of deep learning in CV (computer vision) and NLP (natural language processing) etc., more and more researchers apply deep learning methods to traffic prediction, including CNN [12][13], CNN+LSTM [14] [15] [16] [17][18][19] and GCN [20][21]. The method based on deep learning is more effective than the previous two methods in extracting complex spatial and temporal dependencies.

Although these methods have been used to solve the spatiotemporal prediction problem with good results, we believe that two important aspects have been largely neglected. First of all, when extracting the spatial dependency of data, the previous methods were used directly on the unprocessed global network. Using these methods, a large amount of redundant information may be extracted. Even if there is a good prediction result, it will cost a lot of time and cost. Secondly, when modeling temporal dependency, previous researchers mainly divided the research objects into two types: one is the global road network, and the other is each detector or link. The former will also extract a lot of useless information and may increase the computational cost. The latter separates the relationship between the target location and its highly correlated location and then contextual relationship and located spatial dependency is ignored.

To solve these problems, we proposed a novel path-based, deep learning model: Based on Path CNN-1D + GRU + CNN-2D (P-CGC), to collectively predict traffic flow at every location on the traffic network. We believe that in the whole network, the paths that are frequently used have a strong dependency and we can extract the most useful information for speed prediction from the paths. First, we use Est-matching algorithm [25] to map match the floating car data for extracting the frequently used key path in the road network and divide the detectors’ data on the urban road network according to the key path. Second, CNN and GRU are introduced to model the temporal dependency of loop detectors in each key path. To be specific, CNN-1D fuses the contextual information of the loop detectors in a key path, and GRU is used as a feature extractor to extract the time dimension features. Third, we concatenate the output of every P-CGC and introduce CNN-2D to extract the global spatial dependency of the road network. Finally, a fully connected neural network is used to transform features into the prediction. The main contributions of our paper are summarized as follows:

- We divide the road network into several key paths. Specifically, we use the EST-matching algorithm to map match the floating car data into the map. Then, based on the paths obtained by the map matching algorithm, the global road network is divided into several key paths.
- In extracting the temporal dependency of the data, we introduce the CNN-1D, GRU. Specifically, CNN-1D is used to fuse the contextual information, and GRU is used to extract the features on the time dimension.
- We concatenate the output of all CNN-1D+GRU models and then used CNN to extract the spatial dependency of the data.
- We conduct extensive experiments on Zhangzhou real-world datasets, and the proposed approach achieves a good result.

2. PRELIMINARIES

2.1 Traffic Speed Prediction

There are $N$ loop detectors in the traffic network, and each loop detector generates one average speed value at a time slice, that is, $N$ loop detectors in the road network generate a vector of length $N$ on each time slice. $x_i^t$ denotes the average speed of the detectors $i$ in $t$ time slice. $X_t = (x_1^t, x_2^t, \ldots, x_N^t)$ denotes the average speed of all the detectors (the number of detectors is $N$ in the urban network) in $t$ time slice. $\bar{X} = (X_1, X_2, \ldots, X_T) \in \mathbb{R}^{N \times T}$ denotes the average speed of all the detectors over $T$ time slices. Besides, we set $Y_i^{t \prime}$ to represent the average speed of detectors $i$ in $t \prime$ time slice in the future, and set $Y_{t \prime}$ as the average speed of all the detectors in the future time slice $t \prime$. 


So, the problem of traffic speed prediction is that: given $X_t$ to predict $Y_t = (y_1^t, y_2^t, \ldots, y_N^t) \in \mathbb{R}^{N \times 1}$

### 2.2 Key Path Selection

We believe that there is a strong correlation between the loop detectors in those paths which are used frequently [22][23][24]. This phenomenon is described in Figure 1. In Figure 1, the green route is a key path. A, B, C, D loop detectors are in the same key path while E isn’t in. Intuitively, we believe that the spatiotemporal correlation between A, B, C, D is strong. Therefore, when we make traffic forecasts, we do not directly consider the data of the whole road network but organize traffic data according to the key paths. Within a key path, more precise and relevant spatiotemporal features can be mined when conducting traffic speed prediction.

In our paper, we use floating car data to obtain key paths of the urban road network. First, we use Est-matching algorithm [25] to match the floating car data into the road network. After map matching, we will obtain a series of detector sequences by look up the link-detectors correspondence table. We excluded detector sequences with less than 3 detectors. Then we extract all the possible detector sequences of the urban road network and make each sequence as a base-sequence. We traverse the detector sequences from map matching and compared them with all base-sequences by Needleman/Wunsch algorithm. The comparison is successful if the weight was greater than the threshold value $\delta$, and we count the number of successful comparisons of each base-sequence. Finally, we choose the base-sequences with the largest number of successful matching until the whole road network is covered. We will apply these detectors sequences grouped by key paths to the next speed prediction.

### 3. P-CGC MODEL

The system architecture of the proposed model P-CGC is shown in Figure 2. First, we use CNN-1D, GRU to extract the temporal dependency of the data. Specifically, CNN-1D is used to fuse the contextual information of the loop detectors which are in the same key path, and GRU is used to extract the features on the time dimension. The number of temporal dependencies capturing models is equal to the number of key paths. After that, we concatenate the outputs of all CNN-1D+GRU models, and then use CNN to capture the global spatial dependency of the data. Finally, a fully connected neural network is used to transform features into the prediction.
3.1 Temporal dependency Modeling

We introduce CNN-1D and GRU to model the temporal dependencies of the detectors which are in the same key path. The model is shown in Figure 3. The number of models is equal to the number of key paths. We believe that this modeling method enables the highly correlated detectors to be concentrated in one model to jointly extract the time correlation. For a key path, we have \( T \) temporal observations \( \mathbf{X} = (X_1, X_2, ..., X_T) \in \mathbb{R}^{N \times T} \) and \( X_t \in \mathbb{R}^{N} \) denotes the \( t \)-th time slice observations of \( N \) detectors. The workflow of temporal dependency modeling is as follows.

Firstly, for a key path, we introduce CNN-1D to fuse the contextual information of the key path and expand the information of the key path to a higher dimensional vector before extracting the temporal dependency of the key path. The contextual information means the information among adjacent detectors. The convolution kernel of CNN-1D moves along the temporal axis. For example, the input of CNN-1D of a key path can be regarded as \( Y \in \mathbb{R}^{C_0 \times T} \). \( \Gamma \in \mathbb{R}^{K \times C_0 \times C_0} \) is the convolution kernel of CNN-1D. \( C_0 \) is the number of channels at a time slice and \( C_0 \) is equal to the number of detectors from one key path in the first CNN-1D layer. \( K \) and \( C_0 \) are the width and number of the convolution kernel, respectively. We use padding to ensure the time dimension the same. The one-dimension convolution with padding can be defined as,

\[
Y^{t+1} = \Gamma^{t} Y^{t} \in \mathbb{R}^{C_0 \times T}
\]

(1)

\( Y^{t} \) represents the input of the current level, and \( Y^{t+1} \) represents the input of the next level. \( \ast \) represents the symbol of CNN-1D. We can stack the CNN-1D layers to capture deep correlation.

Secondly, we use GRU [26] to capture the temporal dependency of the key paths. The principle of GRU is very similar to that of LSTM, which uses gating mechanism to control input, memory, and other information to make predictions at the current timestep. Unlike the LSTM with three gates, GRU has only two gates, namely, a reset gate and an update gate. Compared with LSTM, GRU can also achieve a considerable effect. And because the gates number of GRU is one less than LSTM per cell, the number of GRU parameters is smaller than LSTM with all other hyper-parameter being equal, which indicates that GRU is easier to train than LSTM. Therefore, we prefer to use GRU. In our paper, the input of the GRU is \( Y \in \mathbb{R}^{C_0 \times T} \), which is the output of the last CNN-1D layer. We split the input into \( T \) pieces and input them into each GRU cell separately. The final output of the GRU for each key path is a vector and it contains the contextual information and the time dimension information of the key path. We can stack the GRU layers to capture deep correlation like CNN-1D. GRU can be defined as,

\[
h_t = \text{GRU}(x_t, h_{t-1})
\]

(2)

where \( x_t \in \mathbb{R}^{C_0 \times 1} \) is the input of the \( t \) th cell, \( h_{t-1} \) is the hidden value of \( (t-1) \) th cell and \( h_t \) is the hidden value of \( t \) th cell. The internal structure of GRU is shown as following,

\[
z_t = \sigma(W_z x_t + U_z h_{t-1})
\]

(3)

\[
r_t = \sigma(W_r x_t + U_r h_{t-1})
\]

(4)

\[
h_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}))
\]

(5)

Figure 2. System architecture of the proposed P-CGC.
\[
    h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t
\]

update gate \( z_t \) defines how much of the previous memory to keep around. Reset gate \( r_t \) determines how to combine the new input with the previous memory. \( \tilde{h}_t \) is cell value and \( h_t \) is hidden value.

Finally, a fully connected neural layer is used to transform the vector’s features to make sure the output dimension of each model is the same.

### 3.2 Spatial dependency Modeling

After capturing the temporal dependency, we need to further extract the global spatial correlation. We concatenate all the output vectors of the temporal dependency capturing model and form a \( N \times C \) matrix, where \( N \) represents the number of key paths, and \( C \) represents the output dimension of the fully connected neural layer connected after the GRU. We capture spatial dependency by stacking CNN layers. If the dimension of the matrix is not very large, we cannot use pooling. The two-dimension convolution without pooling can be defined as,

\[
    Z_t^{l+1} = \Gamma *_L Z_t^l
\]

Where \( Z_t^l \in \mathbb{R}^{N \times C_l}, Z_t^{l+1} \in \mathbb{R}^{N \times C_{l+1}}, \Gamma \in \mathbb{R}^{m \times m \times C_{l+1}} \) is the convolutional kernel.

P-CGC can be trained via back-propagation by minimizing the mean absolute error (MAE) between predicted values and ground truths.

## 4. EXPERIMENTS

### 4.1 Datasets

The loop detectors data and floating car data of the experiments were collected from the urban area of Zhangzhou, Fujian. The speed data was collected through the loop detectors on the road. Data from 20 loop detectors were selected for this study. The period was one month from May 1st to May 30th in 2017 and the time interval was 5 minutes. So, the time of a day is divided into 288 time periods. In this study, the data are mainly divided into two parts. In the first part, the model is trained from the 1st to the 24th in 2017, where 80% of the models are randomly selected to create the training set, and the rest are used for validation. The second part is the test set from 25th to 30th in 2017. Also, we collected the floating car data from April 23rd to April 29th in 2017 for key paths selection.

### 4.2 Key Paths Selection

Before building the model, we first look for the key path and assign loop detectors to the key paths we have selected. We obtained 5817 paths from floating vehicle data in Zhangzhou. According to the key path selection algorithm in the previous paragraph, we selected 7 key paths for the study area. Therefore, there will be 7 temporal deep learning models in our study.
4.3 Experimental Setting
Recall that the learning task is formulated as learning a function \( f: \mathbb{R}^{N \times T} \rightarrow \mathbb{R}^{N} \). In the experiment, all CNN-1D models have 2 layers with 32 convolution kernels. All GRU model have 2 layers, where the size of hidden neurons in GRU is 32. The fully-connected layers behind the GRU model have 1 layer with 32 hidden neurons. The CNN-2D model has 5 layers and each layer has 64 convolution kernels. All convolution layers aren’t be set pooling layer. Batch size equals 32 and epoch equals to 20. We choose Adam optimizer to optimize our model.

We compare the proposed model (P-CGC) with the following methods for traffic speed forecasting:
- HA, which models the traffic speed prediction as a temporal process, and uses all detectors’ average speed of the previous 25 minutes to predict the next 5 minutes.
- ARIMA, which use one detector average speed of the previous 25 minutes to predict the next 5 minutes.
- CNN, which models the speed in the network as a matrix and used CNN to predict, which is similar to that proposed by Ma et al.,
- LSTM, which uses all detectors average speed of the previous 25 minutes to predict the next 5 minutes,
- GRU, which is similar to the LSTM model,
- CGC, whose structure is the same as P-CGC except that the method of the key path is not used.

4.4 Performance comparison
We compare our models with the eight methods on the Zhangzhou dataset. Table 1 shows the average results of traffic speed prediction performance over the next 5 minutes. We evaluate the performance based on two popular metrics, i.e., Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

We observe the following phenomena from the experiments: (1) deep learning-based methods, including CNN, LSTM, GRU, CGC, P-CGC, which can model the non-linear spatiotemporal dependency, generally outperform other methods, such as HA and ARIMA. This phenomenon indicates that the model based on deep learning can better extract the complex temporal correlation of traffic network; (2) P-CGC achieves the best performance and only the simple CNN takes less time than our model, which suggests the effectiveness of proposed approaches for spatiotemporal correlations modeling; (3) the model with the key path is a little better than the one without key-path. The reason it doesn't do much better is that our road network is small.

| Model  | MAE  | RMSE | Time/s |
|--------|------|------|--------|
| HA     | 0.182| 0.537|        |
| ARIMA  | 0.178| 0.369|        |
| CNN    | 0.055| 0.072| 67     |
| LSTM   | 0.061| 0.089| 336    |
| GRU    | 0.086| 0.101| 321    |
| CGC    | 0.058| 0.075| 235    |
| P-CGC  | 0.054| 0.071| 242    |
4.5 Effect of spatial and temporal dependency modeling

To investigate the effect of spatiotemporal dependency modeling, we evaluate the following variants of P-CGC by removing different components from the model or change the model, including (1) removing CNN-1D, (2) changing LSTM to GRU, (3) removing CNN-2D. The results are shown in Table 2. We observe the following phenomena:

Table 2: Effect of spatial and temporal dependency modeling

| Model                | MAE  | RMSE | Time/s |
|----------------------|------|------|--------|
| removing CNN-1D      | 0.056| 0.074| 305    |
| changing LSTM to GRU| 0.062| 0.080| 267    |
| removing CNN-2D      | 0.058| 0.077| 149    |
| P-CGC                | 0.054| 0.071| 242    |

- The result of “removing CNN-1D” model is worse than P-CGC, maybe in our dataset, the contextual information and located spatial dependency of the key path is important. And CNN-1D can capture the information of key path before extracting the time dependency, so it achieves a good effect.

Figure 4. The predicted value and the real value of entire road network detectors in ten time slices in a day. The predicted value is shown above, and the real value is shown below.
The results of GRU is better than that of the LSTM model, the reason is that GRU has fewer parameters, so it is easier to train and more suitable for us.

The model without CNN-2D doesn’t behave well. The reason may be that we capture the located dependency of each key path using CNN-1D before, but the global spatial dependency is never captured. CNN-2D can handle this problem. So, the model without CNN-2D cannot capture this correlation.

4.6 Results Visualization
To intuitively reflect the effect of the model, we randomly selected two periods (one hour) to visualize the changes of the predicted value and the ground truth of the detectors of the entire road network. Visualization is shown in Figure 4.

From Figure 4, we can conclude that our model better captures the spatiotemporal distribution of speed characteristics. But the result of the prediction was too smooth, so our model cannot perform well in the peak prediction.

5. CONCLUSION
In this paper, we investigated the traffic speed prediction problem and identified its unique spatiotemporal correlations. We proposed a novel deep learning-based model named P-CGC. We divide the road network into several key paths which are based on floating car data and build the model for each key path. We introduce the CNN-1D, GRU in extracting the temporal dependency of the data. Then we concatenate the output of all CNN-1D+GRU models and use CNN to extract the global spatial dependency of the data. Finally, a fully connected neural network is used to transform features into the prediction. When evaluated Zhangzhou real-world datasets, and the proposed approach achieves a better result than other models. Due to the limitation of the dataset, our experimental network is small. In the future, we plan to study in a larger and more complex network and improve key path selection methods and criteria.

ACKNOWLEDGMENT
This work was jointly supported by the National Natural Science Foundation of China, which provided support for the projects entitled “Connected vehicle big data driven expressway multi-objective coordinated control fusing deep learning and traffic flow model (award number 71901070).

REFERENCES
[1] X. Y. Xu, J. Liu, H. Y. Li, and J. Q. Hu, “Analysis of subway station capacity with the use of queueing theory,” Transportation Research Part C Emerging Technologies, vol. 38, no. 1, pp. 28–43, Jan. 2014.
[2] P. Wei, Y. Cao, and D. Sun, Total unimodularity and decomposition method for large-scale air traffic cell transmission model, Transportation Research Part B, vol. 53, no. 3, pp. 1 16, Jul. 2013.
[3] W. Qi, L. I. Li, H. U. Jianming, and B. Zou, Traffic velocity distributions for different spacings, Journal of Tsinghua University, vol. 51, no. 3, pp. 309 312, Mar. 2011.
[4] F. F. Xu, Z. C. He, and Z. R. Sha, “Impacts of traffic management measures on urban network microscopic fundamental diagram,” Journal of Transportation Systems Engineering and Information Technology, vol. 13, no. 2, pp. 185–190, Apr. 2013.
[5] M. S. Ahmed and A. R. Cook, “Analysis of freeway traffic timeseries data by using box-jenkins techniques,” Transportation Research Board. no. 722, pp. 1-9, 1979.
[6] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, “Shortterm prediction of traffic volume in urban arterials,” Journal of Transportation Engineering, vol. 121, no. 3, pp. 249–254, 1995.
[7] I. Okutani and Y. J. Stephanedes, “Dynamic prediction of traffic volume through kalman filtering theory,” Transportation Research.
[8] C. P. I. J. V. Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. V. Lint, and H. J. V. Zuylen, “Localized extended kalman filter for scalable real-time traffic state estimation,” IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 1, pp. 385–394, Mar. 2012.

[9] L. L. Ojeda, A. Y. Kibangou, and C. C. de Wi, “Adaptive Kalman filtering for multi-step ahead traffic flow prediction,” 2013 American Control Conference, Aug. 2013.

[10] Van Lint, J., and Van Hinsbergen, C. 2012. Short-term traffic and travel time prediction models. Artificial Intelligence Applications to Critical Transportation Issues 22(1):22–41.

[11] Jeong, Y.-S.; Byon, Y.-J.; Castro-Neto, M. M.; and Easa, S. M. 2013. Supervised weighting-online learning algorithm for short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems 14(4):1700–1707.

[12] Xiaolei M, Zhuang D, Zhengbing H, et al. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction[J]. Sensors, 2017, 17(4).

[13] Zhang J, Zheng Y, Qi D, et al. Predicting Citywide Crowd Flows Using Deep Spatio-Temporal Residual Networks[J]. Artificial Intelligence, 2017, 259.

[14] Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; and Ye, J. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In AAAI Conference on Artificial Intelligence, 2588–2595

[15] Haiyang Y, Zhihai W, Shuqin W, et al. Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks[J]. Sensors, 2017, 17(17):1501.

[16] Liu Y, Zheng H, Feng X, et al. Short-term traffic flow prediction with Conv-LSTM[C]// 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2017.

[17] Liu L, Qiu Z, Li G, et al. Contextualized Spatial-Temporal Network for Taxi Origin-Destination Demand Prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, PP(99):1-13.

[18] Yao H, Tang X, Wei H, et al. Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33:5668-5675.

[19] Zhou X, Shen Y, Zhu Y, et al. Predicting Multi-step Citywide Passenger Demands Using Attention-based Neural Networks[C]// the Eleventh ACM International Conference. ACM, 2018.

[20] Yu B, Yin H, Zhu Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting[C]// Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18. 2018.

[21] Zhao L, Song Y, Zhang C, et al. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, PP(99):1-11.

[22] De Fabritiis, C., Ragona, R., Valenti, G., 2008. Traffic estimation and prediction based on real time floating car data. In: International IEEE Conference on Intelligent Transportation Systems, pp. 197–203.

[23] Deng, M., Qu, S., 2016. Road short-term travel time prediction method based on flow spatial distribution and the relations. Mathematical Problems in Engineering 2016.

[24] Wang J, Chen R, He Z. Traffic speed prediction for urban transportation network: A path based deep learning approach[J]. Transportation Research Part C: Emerging Technologies, 2019, 100(MAR.):372-385.

[25] LIU, Z., FANG, J., XU, M., et al. Vehicle Trajectory Enhanced Map Matching Method for Low Frequency GPS Data. The 20th COTA. International Conference of Transportation Professionals, 2020.

[26] Chung J, Gulcehre C, Cho K H, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. Eprint Arxiv, 2014