Multistep Flow Prediction on Car-Sharing Systems: A Multi-Graph Convolutional Neural Network with Attention Mechanism

Yi Luo*[†], Qin Liu*[†‡], Hongming Zhu[†‡], Hongfei Fan[†‡], Tianyou Song[†], Chang Wu Yu[§] and Bowen Du[¶]
[†]School of Software Engineering, Tongji University
[‡]Tsingtao Advanced Research Institute, Tongji University
[§]Department of Computer Science and Information Engineering, Chung Hua University
[¶]Department of Computer Science, University of Warwick
Email: {1731530, qin.liu, zhu_hongming, fanhongfei, 1551177}@tongji.edu.cn, cwyu@chu.edu.tw, B.Du@warwick.ac.uk

***Abstract*—Multistep flow prediction is an essential task for the car-sharing systems. An accurate flow prediction model can help system operators to pre-allocate the cars to meet the demand of users. However, this task is challenging due to the complex spatial and temporal relations among stations. Existing works only considered temporal relations (e.g., using LSTM) or spatial relations (e.g., using CNN) independently. In this paper, we propose an attention multi-graph convolutional sequence-to-sequence model (AMGC-Seq2Seq), which is a novel deep learning model for multistep flow prediction. The proposed model uses the encoder-decoder architecture, wherein the encoder part, spatial and temporal relations are encoded simultaneously. Then the encoded information is passed to the decoder to generate multistep outputs. In this work, specific multiple graphs are constructed to reflect spatial relations from different aspects, and we model them by using the proposed multi-graph convolution. Attention mechanism is also used to capture the important relations from previous information. Experiments on a large-scale real-world car-sharing dataset demonstrate the effectiveness of our approach over state-of-the-art methods.**

***Index Terms*—Car-sharing systems, Multistep flow prediction, Graph convolution network**

I. Introduction

In recent years, car-sharing systems have been introduced to a number of cities as a means of increasing mobility, reducing congestion, and pollution [1]. Car-sharing systems involve a small to medium fleet of cars, which are available at several stations, to be used by a relatively large group of users. Users can pick up a car in a station and drop off it at another station, which is called the one-way system [2]. The key to success of the car-sharing systems is an accurate flow prediction model, which plays a vital role in various tasks such as car rebalancing [3]. Furthermore, instead of predicting only for the next step (e.g., next day), the multistep flow prediction is more attractive to the system operators since it offers information on a long-term trend.

Traditional time series prediction methods like ARIMA have been widely used for traffic prediction problem [4, 5]. However, these approaches are often applied to a single station separately and ignore the spatial relations with each other. For example, if a station near a railway station has high flow, another one close to it may also have high flow. Furthermore, building a separate prediction model for each station is time-consuming and impractical if there are hundreds of stations. Hence, the key challenge for this problem lies in how to model complex spatial relations and temporal dynamics. To tackle the above challenges, we propose a novel deep learning model, named **a**ttention **m**ulti-**g**raph **c**onvolutional **seq**uence-**to-seq**uence model (AMGC-Seq2Seq), which captures spatial-temporal relations effectively for station-level flow prediction.

In this work, a new multistep flow prediction model for car-sharing systems is designed. Multiple graphs among stations are defined to represent their heterogeneous spatial relations. Then we employ the proposed multi-graph convolution to model these spatial correlations. Furthermore, a novel deep learning framework, named AMGC-Seq2Seq, is proposed to capture the spatial and temporal relations simultaneously by incorporating the encoder-decoder architecture with graph convolution networks. The proposed method is validated on a large-scale real-world car-sharing dataset from EVCARD. The dataset contains car orders through EVCARD service in the city of Shanghai in China over three months, with about 480,000 orders per month on average. We conducted extensive experiments to compare with state-of-the-art methods and have demonstrated the superior performance of our proposed method.

The rest of this paper is organized as follows: Section II reviews the existing works. Section III first formulates the multistep flow prediction problem and then describes the details of the proposed AMGC-Seq2Seq model. Section IV presents the experiment settings and discusses the obtained results. Section V concludes this work.

II. Related Work

The problem of car-sharing system flow prediction is similar to traffic prediction problem, of which the goal is to predict the traffic-related value (e.g., traffic flow or traffic speed) for a period of time through historical data. A number of studies

*These authors contributed equally to this work

DOI reference number: 10.18293/SEKE2019-051

have investigated traffic prediction for decades. In this section, we discuss the related work on traffic prediction problem.

The early research on traffic prediction focused on the prediction of the individual station using classical empirical statistical methods. Among all the traditional methods, the autoregressive integrated moving average (ARIMA) and its variants are the most widely used [6]. Based on this time series model, recent studies also consider adding external context data, such as weather, wind speed and event information [7]. Besides, various techniques have been used to model spatial interactions. Deng et al. [8] apply non-negative matrix factorization on road networks to capture correlations between roads. Tong et al. [9] mainly adopt POI data as the spatial features. However, all of these methods are based on the time series model and ignore complex spatial-temporal relations.

With the success in deep learning, more and more researchers attempt to use deep learning techniques on traffic prediction problem. Zhao et al. [10] use the long short-term memory (LSTM) networks to capture non-linear temporal relations. Wang et al. [11] propose the DeepSD, which utilizes multiple data sources and predict the gap between the car-hailing supply and demand. These methods focus on temporal features extraction but do not model the spatial-temporal relations.

To effectively model the complex spatial relations, some researchers use convolutional neural network (CNN) to capture adjacent relations among the traffic networks. Yao et al. [12] propose the DMVST-NET, which models both spatial and temporal relations by local CNN and LSTM. Since the traffic networks are naturally non-Euclidean as the data format is no longer a matrix and CNN becomes less helpful, some researchers turn to use graph convolutional network (GCN) to model this non-Euclidean structures. Chai et al. [13] propose a multi-graph convolutional network to catch heterogeneous inter-station spatial correlations. However, all of these methods are proposed for one-step prediction. Specifically, for multistep prediction, the output from the previous step is taken as the input to the current step, which usually leads to error accumulation and poor prediction performance.

Several researchers have recently attempted to investigate multistep prediction. Cai et al. [14] propose an improved KNN model to achieve multistep forcasting. They describe the traffic state of a road segment by a spatial-temporal state matrix and use the Gaussian weighted Euclidean distance to measure the similarity. Park et al. [15] propose the AGC-Seq2Seq for multistep speed prediction, which learns the spatial-temporal relations simultaneously by integrating LSTM and GCN. They utilize encoder-decoder to model the multistep prediction problem. However, they use only one graph to model the spatial relations, which may not be enough to reflect the complex spatial relations, and they have not capture the local temporal relations for the individual station.

Inspired by those research accomplishments and the real-world problem observations, a potential solution for more accurate and practical prediction should be an integrated analysis for both spatial and temporal relations of stations.

III. Proposed Model

In this section, we first formulate the problem and then describe how to model the spatial and temporal relations using the proposed attention multi-graph convolutional sequence-to-sequence model (AMGC-Seq2Seq).

A. Problem Formulation

In car-sharing systems, there are two types of flows: outflow and inflow. The outflow of station i is defined as the pick-up frequency at the time slot t (e.g. one day), which is denoted by $y_{i,t}^{out}$. The inflow of station i is defined as the drop-off frequency at the time slot t, which is denoted by $y_{i,t}^{in}$.

Suppose we have N stations, then the outflow of all stations at time slot t can be denoted as $Y_t^{out} = [y_{1,t}^{out}, y_{2,t}^{out}, \dots, y_{N,t}^{out}]$, and inflow of all stations at time slot t can be denoted as $Y_t^{in} = [y_{1,t}^{in}, y_{2,t}^{in}, \dots, y_{N,t}^{in}]$.

Suppose the current time slot is t, and we have the historical data $[(Y_1^{in}, Y_1^{out}), (Y_2^{in}, Y_2^{out}), \dots, (Y_t^{in}, Y_t^{out})]$, the problem considered in this work is to predict the flow at next T steps $[(\hat{Y}_{t+1}^{in}, \hat{Y}_{t+1}^{out}), (\hat{Y}_{t+2}^{in}, \hat{Y}_{t+2}^{out}), \dots, (\hat{Y}_{t+T}^{in}, \hat{Y}_{t+T}^{out})]$, aiming to:

$$min \sum_{k=1}^{T} \left\| \hat{Y}_{t+k}^{in} - Y_{t+k}^{in} \right\|_2^2, min \sum_{k=1}^{T} \left\| \hat{Y}_{t+k}^{out} - Y_{t+k}^{out} \right\|_2^2 \quad (1)$$

B. Framework Overview

Figure 1 shows the architecture of our proposed model. Generally, our model uses the encoder-decoder architecture. In the encoder, we form a ”hamburger” structure with two LSTMs and one multi-graph convolution layer(M-GCN) in between to model the spatial and temporal relations. First, LSTM is used to model local temporal information for each station. After that, multi-graph convolution is used to model heterogeneous spatial realtions among stations. Finally, another LSTM is used to aggregate spatial-temporal relations together. Then the encoded information is passed to the decoder and incorporated with attention mechanism to generate multistep outputs. The details of each module are described as follows.

C. Encoder

1) Temporal relations modeling: Since the flow pattern for each station varies a lot, we adopt Long Short-Term Memory (LSTM) network [16] to model this local temporal relation. At each time step t, LSTM takes two inputs: memory of the last time step h_{t-1} and the related information at current time step x_t. Based on these inputs, LSTM learns to remove or add new information to the memory, and finally generates a new memory state h_t which accumulates all the previous information. This process is controlled by three gates: forget gate, input gate, and the output gate, which can be formulated as follows:

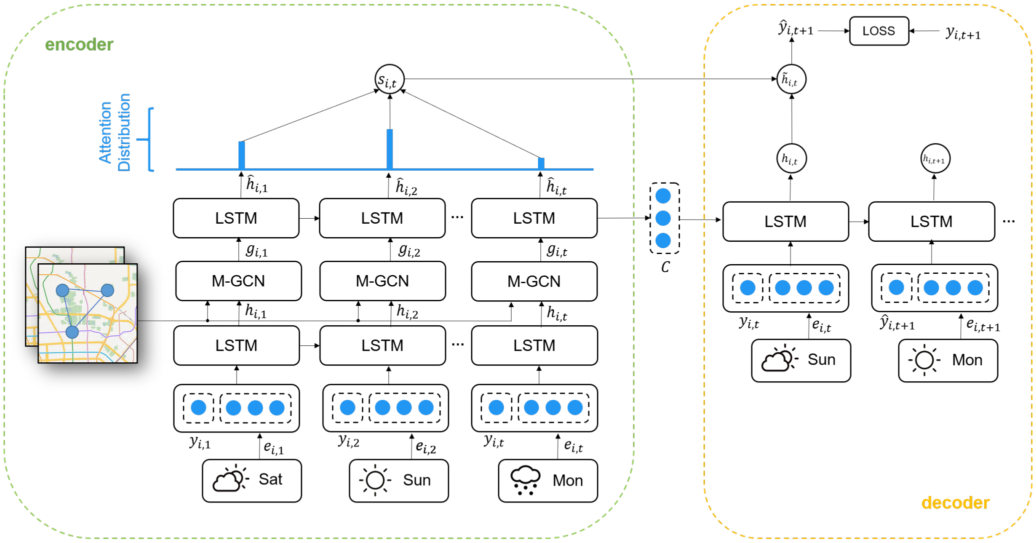

Fig. 1. The architecture of AMGC-Seq2Seq.

$$f_{i,t} = \sigma(W_f[h_{i,t-1}, x_{i,t}] + b_f) \tag{2}$$

$$i_{i,t} = \sigma(W_i[h_{i,t-1}, x_{i,t}] + b_i) \tag{3}$$

$$\widetilde{C}_{i,t} = tanh(W_C[h_{i,t-1}, x_{i,t}] + b_c) \tag{4}$$

$$C_{i,t} = f_{i,t} \circ C_{i,t-1} + i \circ \widetilde{C}_{i,t} \tag{5}$$

$$o_{i,t} = \sigma(W_o[h_{i,t-1}, x_{i,t}] + b_o) \tag{6}$$

$$h_{i,t} = o_{i,t} \circ tanh(C_{i,t}) \tag{7}$$

Where $\circ$ denotes the Hadamard product. $f_{i,t}$, $i_{i,t}$ and $o_{i,t}$ are the forget gate, the input gate and the output gate respectively. σ and $tanh$ are the nonlinear activation functions. W_f, W_i and W_o are all trainable parameters, while b_f, b_i and b_o are the corresponding bias vectors.

As for the inputs of step t, we concatenate flow $y_{i,t}$ with external features $e_{i,t}$ (e.g., weather, weekday/weekend) together:

$$x_{i,t} = y_{i,t} \oplus e_{i,t} \tag{8}$$

It should be noted that all stations share the same weights of LSTM in the proposed model. The reason is that sharing LSTM among all stations may encourage the desired model becomes more general and reduce complexity.

2) Spatial relations modeling: To capture the spatial relations between stations, we propose a multi-graph convolution layer. The goal of the multi-graph convolution layer is to learn a function of features on graphs. Here we define the graphs which will be used later.

The car-sharing systems can be modeled as a weighted undirected graph, of which a node represents one station and an edge represents the relation between two stations. Usually, the large the weight of an edge is, the strong correlations there are between two stations. The simplest graph is the distance graph, where the weight of an edge is defined as the reciprocal of the distance. In addition to the distance graph, there can be more graphs used to model the relation of stations. In this work, we propose and define the following two graphs: distance graph and POI graph.

Distance Graph: According to our observation, stations in the same area are likely to have similar flows. Therefore, the edge in the distance graph between two stations is defined to be the reciprocal of the distance.

$$A_{dis,i,j} = \begin{cases} {d_{i,j}}^{-1}, & i \neq j \\ 0, & i = j \end{cases} \tag{9}$$

Where $d_{i,j}$ is the distance between station i and j.

POI Graph: Intuitively, stations sharing similar functionality may have similar flows. For example, stations in the commercial area usually have more flows on weekends, and stations in the office area are expected to have more flows on weekdays. Point of interest, or POI, is a specific point location that someone may find useful or interesting(e.g., schools, shops, post offices are all POIs). Therefore, we can use POIs around a station to represent its functionality. Accordingly, we define the edge in a POI graph between two stations as the cosine similarity of POIs.

$$A_{dis,i,j} = \begin{cases} \frac{P_i \cdot P_j}{\|P_i\|\|P_j\|}, & i \neq j \\ 0, & i = j \end{cases} \tag{10}$$

Here P_i and P_j are the POI vectors of station i and j respectively. The dimension of the vector is the category of POIs, and the value in the vector is the number of the specific POI category around the station.

Recall that in the temporal modeling, we have the hidden state $h_{i,t}$, which contains the temporal information of station i at t time step. Here we define $H_t = [h_t^1, h_t^2, ..., h_t^N]$, which represents the temporal information of all stations at the time t. Then with the above graphs constructed, we propose the multi-graph convolution to model the spatial relations as defined in eq.(11).

$$G_t = \sigma(\sum_{A \in \mathbb{A}} A * H_t * W) \tag{11}$$

Where $\mathbb{A}$ is the set of graphs, H is the feature matrix of all stations, and W is a trainable matrix which will be updated during the training. Here σ is a non-linear activation function, which is ReLU in our model.

Eq.(11) indicates that for each station, we update its feature by a weighted sum of the features of all the other stations. The larger the weight of the edge is between two stations, the more the feature of that station contributes. However, there are some problems need to be resolved.

First of all, according to the definition of the graph, the diagonal of the graph matrix contains all zeros. As a result, if we multiply it with the feature matrix, the vector of itself contributes nothing, which loses a lot of important information of itself. We fix this by adding an identity matrix to A.

Another problem is that since we combine multiple graphs by adding the transformed feature matrices together, the obtained graph matrices may vary a lot. Hence, we normalize A by dividing each value by the row sum such that all rows sum to one.

Therefore, we modify the eq.(11) as follows:

$$D_{i,j} = \begin{cases} \sum\limits_{k=1}^{N} A_{i,k}, & i = j \\ 0, & i \neq j \end{cases} \tag{12}$$

$$\hat{A} = D^{-1}A + I \tag{13}$$

$$G_t = \sigma(\sum_{\hat{A} \in \hat{\mathbb{A}}} \hat{A} * H_t * W) \tag{14}$$

Where D is a diagonal matrix of which the value in the main diagonal is the row sum of A. Multiplying D^{-1} with A makes all rows of A sum to one. Finally, we add it with the identity matrix I to ensure self-loops in the graph.

Finally, another LSTM is applied to each station to aggregate both temporal and spatial relations for station i. The final hidden state $h_{i,t}$ is selected as the context vector c_i for station i which stores all the information of the encoding, and then this vector is passed to the decoder as the initial state to be decoded as shown below.

$$\hat{h}_{i,t} = LSTM(\hat{h}_{i,t-1}, g_{i,t}) \tag{15}$$

$$c_i = \hat{h}_{i,T'} \tag{16}$$

D. Decoder

In the decoder, a separate LSTM is used to decode context vector c_i to obtain the multistep outputs. The LSTM part is the same as the equations (2)-(7), while the initial state is set as the context vector which stores all the information of previous time steps.

$$h_{i,0} = c_i, \tag{17}$$

$$h_{i,t} = LSTM(\hat{h}_{i,t-1}, x_{i,t}) \qquad t > 0 \tag{18}$$

Furthermore, we employ the attention mechanism which is widely used in most of NLP scenarios [17]. At a high-level, an attention mechanism enables our neural network to focus on relevant parts of the inputs more than the irrelevant parts when performing a prediction task. For example, if the current time step is Sunday then the information of the Sunday before a week are considered as much help to predict the current output. Let t and $t^{'}$ denote the time step at decoder and encoder respectively, and the attention mechanism works as follows:

$$u_{i,t}^{t^{'}} = q_a^T tanh(W_a[h_{i,t} + \hat{h}_{i,t'}]) \tag{19}$$

$$a_{i,t}^{t^{'}} = \frac{exp(u_{i,t}^{t^{'}})}{\sum_{t^{'}=1}^{T^{'}} exp(u_{i,t'})} \tag{20}$$

$$s_{i,t} = \sum\nolimits_{t^{'}=1}^{T^{'}} a_{i,t}^{t^{'}} \hat{h}_{i,t'} \tag{21}$$

Where the weight $a_{i,t}^{t^{'}}$ measures the importance of the time step $t^{'}$ in t. Here $a_{i,t}^{t^{'}}$ is derived by comparing the current hidden state $h_{i,t}$ with the previous spatial-temporal hidden state $\hat{h}_{i,t'}$. The attention vector s_{i_t} is a weighted sum of hidden states in each previous time step $t^{'}$.

Finally, we concatenate attention vector s_i with current hidden state h_i as $h_i^{'}$. Then we feed $h_i^{'}$ to a fully connected layer and get the final prediction. Noted that in this work, we predict inflow and outflow simultaneously.

$$\tilde{h}_{i,t} = h_{i,t} \oplus s_{i,t} \tag{22}$$

$$[\hat{y}_{i,t}^{in}, \hat{y}_{i,t}^{out}] = W_y \tilde{h}_{i,t} + b_y \tag{23}$$

Since this is a multistep problem, the loss function is defined as the mean squared error:

$$loss = \frac{1}{T} \sum_{t=1}^{T} (y_{i,t}^{in} - \hat{y}_{i,t}^{in})^2 + (y_{i,t}^{out} - \hat{y}_{i,t}^{out})^2 \tag{24}$$

IV. Experiment

A. Dataset

We use a large-scale car-sharing dataset provided by EVCARD, which is one of the biggest hourly rental operators in China. The dataset contains the orders from 5/1/2017 to 7/31/2017 in Shanghai, and there are about 15,000 orders per day. The order includes pick-up station, drop-off station, pick-up time and drop-off time. Weather data is collected from JUHE[1] website. POI data is collected through AMap API[2],

[1] https://www.juhe.cn/docs/api/id/277

[2] https://lbs.amap.com/

which contains 15 primary categories. For each station, we collected POIs within $1km$ around it and represented them in a vector, whose entry is the number of a specific POI category. We summarize the statistics of the dataset in Table I.

The data from 5/1/2017 to 6/30/2017 are used for training (61 days), and the data from 7/1/2017 to 7/31/2017 are used for testing (31 days).

TABLE I
DATASET STATISTICS

Data Source	**EVCARD**		
Time from	5/1/17	7/1/17	
to	6/30/17	7/31/17	
#days	61	31	
#stations	1433	1433	
#orders	964,531	498,856	
Data Source	**AMap API**		
POI type	number	POI type	number
food	520,779	shopping	1,065,138
life service	550,200	sports	103,723
medical service	80,934	accommodation	56,490
tourist	11,773	residence	217,018
government	138,298	education	161,613
transportation	248,822	finance service	81,673
enterprises	455,720		

B. Experiment settings

In the experiment, we use the past 14-day historical data to predict the flow in the next 7 days. The number of hidden layers for LSTM is two with 64 hidden units. The dimension of graph convolution is set to 64. Adam [18] is selected as the optimization algorithm, and the initial learning rate is set to 0.001. Here 10% of the training data were selected as the validation set for parameter tuning, and early stopping is used. To speed up convergence, teacher forcing [19] is applied, which means we feed the actual flow to the decoder at the training stage. We perform Xavier initialization to initialize all the trainable parameters. The training process takes about 8 hours on a single TITAN XP GPU.

C. Baseline & Metric

We compare the proposed model (AMGC-Seq2Seq) with the following methods:

- **Historical Average** (HA): The historical average model predicts the flow by using the average value of history. In our experiment, the prediction is the average from the same time in previous weeks.
- **Autoregressive Integrated Moving Average** (ARIMA): ARIMA is a widely used time series prediction model. There are three parameters (p, d, q) need to be set for the model. The degree of differencing is set as $d = 1$. Here p and q are determined by grid search on the training set.
- **Seq2Seq**: Sequence to sequence model has been proved to be effective for time series prediction problem. Same as our model, Seq2Seq model is trained on all stations.
- **AGC-Seq2Seq [15]**: AGC-Seq2Seq is a graph based seq2seq model to predict traffic speeds. The encoder of that model is different from ours in two aspects: (1) They employ only one graph while we combine multi-graphs, and (2) we have one more LSTM which is used to capture the local temporal relations for each station.

We use Mean Average Percentage Error (MAPE) and Rooted Mean Square Error (RMSE) to evaluate the proposed model, which are defined as follows:

$$MAPE = \frac{1}{NT}\sum_{i=1}^{N}\sum_{j=1}^{T}\frac{\left|\hat{y}_{t+j}^{i} - y_{t+j}^{i}\right|}{y_{t+j}^{i}} \tag{25}$$

$$RMSE = \sqrt{\frac{1}{NT}\sum_{i=1}^{N}\sum_{j=1}^{T}(\hat{y}_{t+j}^{i} - y_{t+j}^{i})^2} \tag{26}$$

D. Performance comparison

Table II shows the performance of the proposed model compared to all other competing models. AMGC-Seq2Seq achieves the lowest RMSE (6.15) and the lowest MAPE (23.66) among all the methods. More specifically, HA and ARIMA perform poorly, as they rely on only historical data for prediction. Deep learning methods, including Seq2Seq, AGC-Seq2Seq, and AMGC-Seq2Seq, which are able to model the spatial-temporal relations, generally outperform the traditional methods. Compared with AGC-Seq2Seq, the proposed model further utilizes multi-graph convolution and one more LSTM to capture the local temporal relation for the individual station, which results the lowest RMSE and MAPE.

TABLE II
PERFORMANCE OF DIFFERENT METHODS

Method	RMSE	MAPE(%)
HA	8.03	30.23
ARIMA	7.54	27.89
Seq2Seq	6.73	25.93
AGC-Seq2Seq	6.48	24.82
AMGC-Seq2Seq	**6.15**	**23.66**

E. Effect of multi-graph convolution

Here we study the effect of multi-graph convolution. Table III shows the results when we only use a single graph (distance or POI graph) for prediction. According to the results, we observe that a single graph can be worse than the baseline method(e.g., the model of POI graph yields a result which is worse than Seq2Seq model). However, by combining them, our model beats the AGC-Seq2Seq with the lowest RMSE and MAPE, which proves the effectiveness of the proposed multi-graph convolution.

TABLE III
PERFORMANCE OF DIFFERENT GRAPH

Method	RMSE	MAPE
AGC-Seq2Seq + POI Graph	7.46	30.26
AGC-Seq2Seq + Distance Graph	6.52	25.35
AMGC-Seq2Seq	**6.15**	**23.66**

F. Effect of attention

Figure 2 shows the prediction of station 1 from 7/6 to 7/26, where the data from 7/6 to 7/19 are used as the historical data, and the flow from 7/20 to 7/26 are predicted by our model. In the attention mechanism, $a_{i,t}^{t'}$ in eq.(20) measures the relevance of the historical information in the predicted state. The corresponding attention heatmap of station 1 is depicted in Figure 3, where the darker the color is, the more the relevance there are between two dates. One interesting finding is that when predicting flows on weekdays, the model tends to look at the latest history. However, when predicting flows on weekends (7/22 and 7/23 in this example), the model tends to look at the information on weekends from the history (7/15, 7/16, 7/8 and 7/9 in this example). These results confirm that our model can automatically capture the relevant information to make a more robust prediction.

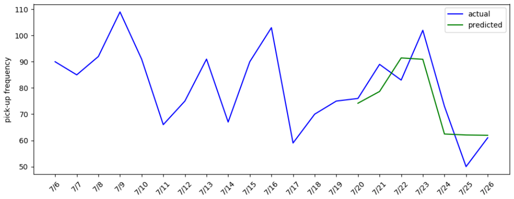

Fig. 2. Outflow of station 1

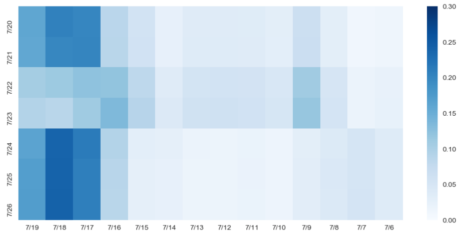

Fig. 3. Attention heatmap of station 1

V. Conclusion

In this paper, we propose a novel deep learning model AMGC-Seq2Seq for flow prediction in a car-sharing system. There are two novelties of the proposed model. The first is that we utilize multi-graph convolution to model the spatial relations from different aspects. The second is that we incorporate LSTMs with a graph convolution network in a "hamburger" structure which capture both spatial and temporal relations effectively. We evaluated the model on a large-scale real-world car-sharing dataset from EVCARD. The experiment results show that the proposed model achieved better results than state-of-the-art baselines. In future, we plan to investigate the following aspects: (1) evaluate the proposed model on other datasets. (2) incorporate more diverse features (e.g., road network) in a car-sharing system.

Acknowledgment

This research is supported by the National Key R&D Program of China under Grant (No.2018YFB0505000, 2018YFB0505000) and the Shanghai Committee of Science and Technology under Grant (No.17511107303, 17511110202). This research is also supported by EVCARD.

References

[1] Richard Katzev. Car sharing: A new approach to urban transportation problems. *Analyses of Social Issues and Public Policy*, 3(1):65–86, 2003.

[2] Angela Febbraro, Nicola Sacco, and Mahnam Saeednia. One-way carsharing: solving the relocation problem. *Transportation Research Record: Journal of the Transportation Research Board*, (2319):113–120, 2012.

[3] Diana Jorge and Gonçalo Correia. Carsharing systems demand estimation and defined operations: a literature review. *European Journal of Transport and Infrastructure Research*, 13(3), 2013.

[4] Billy M Williams and Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003.

[5] Bo Zhou, Dan He, Zhili Sun, and Wee Hock Ng. Network traffic modeling and prediction with arima/garch. In *Proc. of HET-NETs Conference*, pages 1–10, 2005.

[6] Shashank Shekhar and Billy M Williams. Adaptive seasonal time series models for forecasting short-term traffic flow. *Transportation Research Record*, 2024(1):116–125, 2007.

[7] Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. Traffic prediction in a bike-sharing system. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 33, 2015.

[8] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. Latent space model for road networks to predict time-varying traffic. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1525–1534, 2016.

[9] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, Jieping Ye, and Weifeng Lv. The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1653–1662, 2017.

[10] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017.

[11] Dong Wang, Wei Cao, Jian Li, and Jieping Ye. Deepsd: supply-demand prediction for online car-hailing services using deep neural networks. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 243–254, 2017.

[12] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[13] Di Chai, Leye Wang, and Qiang Yang. Bike flow prediction with multi-graph convolutional networks. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 397–400, 2018.

[14] Pinlong Cai, Yunpeng Wang, Guangquan Lu, Peng Chen, Chuan Ding, and Jianping Sun. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C: Emerging Technologies*, 62:21–34, 2016.

[15] Zhengchao Zhang, Meng Li, Xi Lin, Yinhai Wang, and Fang He. Multistep speed prediction on traffic networks: A graph convolutional sequence-to-sequence learning approach with attention mechanism. *arXiv preprint arXiv:1810.10237*, 2018.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.