Crowd Flow Forecasting with Multi-Graph Neural Networks

1st Xu Zhang
Department of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China
zhangx@cqupt.edu.cn

2nd Ruixu Cao
Department of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China
caoruixu@yahoo.com

3rd Zuyu Zhang
Department of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China
changjoey56@gmail.com

4th Ying Xia
Department of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China
xiaying@cqupt.edu.cn

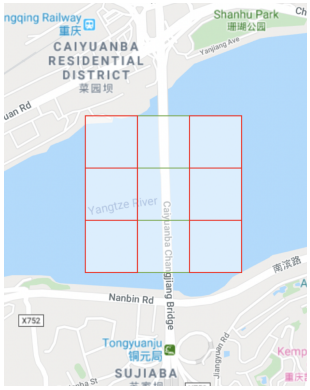
***Abstract*—Crowd flow forecasting is of great significance for urban traffic management and personal travel planning. Due to the complexity of the urban geographic structure and the highly nonlinear temporal and spatial dependence on crowd flow, accurate forecasting becomes very challenging. Recent research works usually divided cities into regions of the same size and coded as heat-maps, for cities with complex terrain, heat-maps contain many invalid data, which have a negative effect on the acquisition of spatial dependence. In order to decrease the effect, we encode the crowd flow into graphs and propose a multi-graph neural network based model to solve the crowd flow forecasting problem. We first construct two K-NN graphs by the Euclidean distance and the Pearson correlation coefficient respectively and the spatial dependence is captured through the spatial block composed of Graph Attention Networks(GAT) and ChebNet, then another ChebNet is deployed to fuse the spatial dependency of the two graphs. Afterward, we adapted a LSTM to capture the temporal dependence of all regions separately and use self-attention mechanism and fully connected layer to get prediction results. Extensive experiment results based on two real-world datasets demonstrate that our model achieves an important performance on other baselines.**

***Index Terms*—crowd forecasting, graph neural networks, attention, data mining**

I. Introduction

Crowd flow forecasting is of great significance for urban traffic management and personal travel planning. Accurate crowd flow forecasting provides city managers a strong basis for traffic decision-making, alleviates traffic congestion, and also helps taxi companies deploying vehicles in advance to save resources and helps individuals plan their trips more effectively.

Crowd flow is affected by two complex and highly nonlinear factors: spatial dependency and temporal dependency. Spatial dependency means that the crowd flow in a certain region is affected by other regions. In addition to neighboring regions, some regions far away may also have a high



(a) the receptive filed of CNN's filter

(b) construction of edges in GCNs

Fig. 1. Difference between regular convolution and graph neural networks

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

correlation on it, e.g. the correlation between office regions and residential regions. Temporal dependency shows that the crowd flow at the next time interval is affected by historical time intervals. Moreover, due to the regularity of human activities, the crowd flow between weekdays shows a certain similarity. Previous studies [1]–[4] divide the cities into grids of the same size, encode the crowd flow into heat-maps and then process it like an image. However, the heat-map contains many invalid regions, such as rivers, lakes, hills in the city, there is seldom human movement in these regions, which denotes zero in heat-map all the time. we call these invalid data. For cities with complex geographical features, a large number of invalid data are not conducive to the acquisition of spatial dependency. As it is shown in figure 1 (a), for predicting the crowd flow in the region where the bridge is located, and if we use a CNN-based approach, the receptive field of CNN's filter is the nine grids, six of which contains invalid data (represented by red border). With the development of graph neural networks, recent studies adapt graph neural networks to solve crowd flow prediction problems. Work [5]–[7] apply graph convolutional networks (GCNs) to highway networks, and naturally, edges are constructed according to the highway networks, this is a relatively straightforward method, however it is not suitable for region-level crowd flow prediction. As it is shown in figure 1(b), we can construct edges between bridge and river bank regions. Moreover, since the two adjacent bridges are functionally alternative, there is a certain interaction between them: when the right bridge is traffic jam, it is foreseeable that people will choose the left bridge, thus we can construct an edge between the two bridges. Actually, we do not need auxiliary information for regional functions, and we get the correlation between regions by analyzing historical data. Research [8] proposes to construct multiple graphs with distances between regions, functional similarities, and highway networks, the disadvantage of this approach is that points of interest and highway networks need to be introduced as auxiliary information.

To address the above problems, the Euclidean distance between regions and the Pearson correlation coefficient of historical data are used as the distance between regions to construct two K-NN unweighted graphs, and then we adapt a graph attention network [9] and Chebyshev network (ChebNet) [10] to capture dynamic spatial , another ChebNet is deployed to fuse two graphs into one. Finally, we use a shared weight LSTM to obtain the temporal dependency of all regions separately and apply a self-attention Mechanism and a fully connected layer to get forecasting results. The contributions of this paper are as follows:

- we propose a novel region-level crowd flow forecasting framework based on graph neural networks and release a new real-world dataset of Chongqing taxi, and our approach outperforms other state-of-the-art models in forecasting accuracy.
- We encode regional-level crowd flow into two K-NN unweighted directed graphs with historical data only.
- We apply a graph attention network and a self-attention mechanism to capture dynamic spatial and temporal correlation, respectively.

II. Related Work

Traffic predictions include forecasts of crowd flow, traffic flow, vehicle speed, taxi demand and other indicators, early traffic prediction research generally use non-deep learning methods, such as time series model like autoregressive integrated moving average (ARIMA) [11] and Kalman filter [12], [13]. These models treat each region independently, however the interaction between regions is ignored. Machine learning models such as K-Nearest Neighbors (KNN) [14], support vector regression (SVR) [15] perform better than statistical models, however the capacity of capturing spatio-temporal connections is still insufficient. Deep learning has shown strong learning and representation capabilities in many fields. In recent years, more and more studies have deployed deep learning models to solve crowd forecasting problem. Zhang et al. [1] utilizes residual neural network (ST-ResNet) to predict crowd flow. Shen et al. [16] propose a model with 3D convolution, which uses three types of convolution kernels to extract features in three dimensions: temporal, spatial, spatio-temporal. Yao et al. [3] first adapt a convolutional neural network to extract the data of each time interval as feature vectors, and then input them into LSTM for time-series modeling. And they propose a spatio-temporal dynamic network [2] for taxi demand prediction which could dynamically learn the correlation between regions. Yuan et al. [17] propose a heterogeneous traffic accident prediction framework based on ConvLSTM, which integrates a variety of auxiliary data, they divide the target region into three types: urban, rural, and mixed, and train different models for different types of regions to address the spatial heterogeneity problem. CNN-based models can effectively extract the features of grid data but cannot be applied to non-Euclidean data. Graph neural network solves this problem, in recent years, graph neural network-based traffic prediction models have received increasing attention. Li et al. [7] propose a DCRNN model, utilizing a bidirectional random walk on the graph and an encoder-decoder architecture with scheduled sampling to capture spatial and temporal correlations, respectively. Yu et al. [6] design the STGCN model consisting of two spatio-temporal convolution module which uses two kinds of graph convolutions, GCN and ChebNet, and one-dimensional CNN to model spatio-temporal correlations, respectively. Geng et al. [8] construct three graphs from three aspects: connectivity, positional relationship, and functional similarity between regions with auxiliary information, they extract the spatio-temporal dependency and get the sub-result of each graph, the final prediction results are calculated as the average of each sub-result.

III. Methodology

A. Problem Definition

We divide the city into N regions and each region can be the same or not, in this article we consider the entire city, in

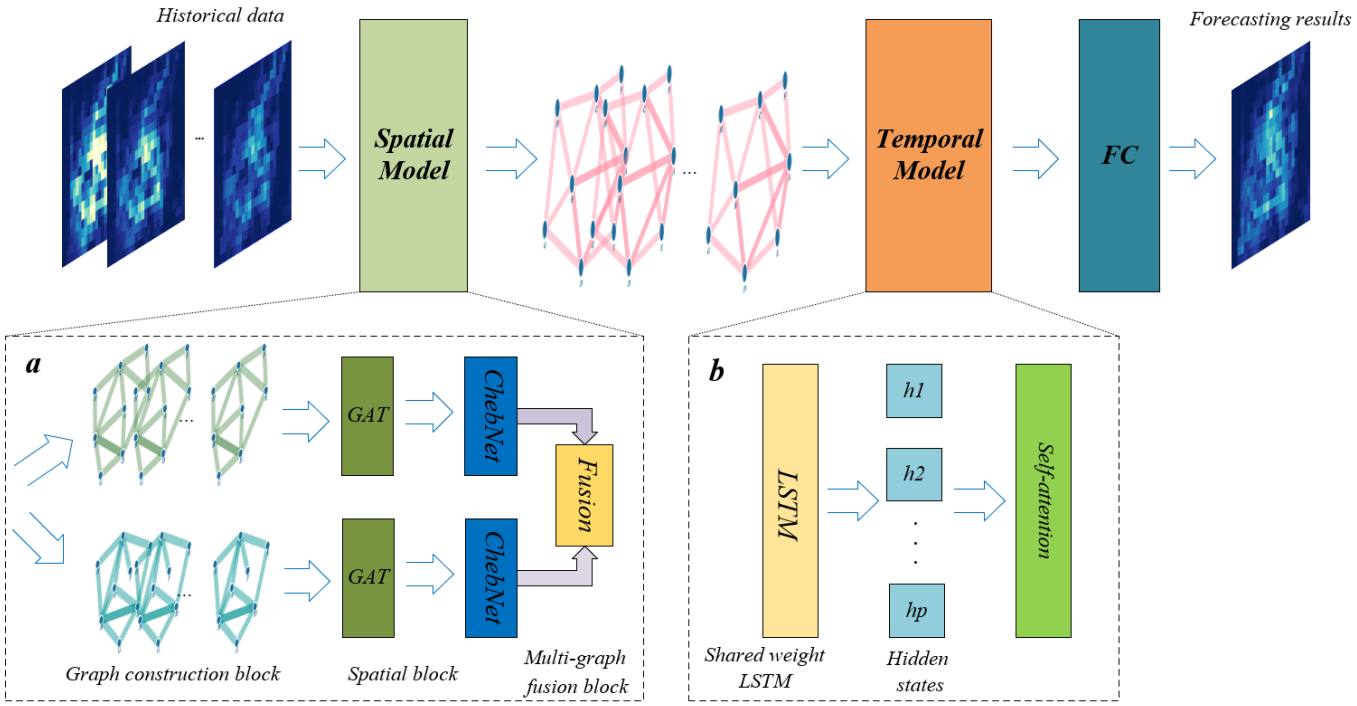


Fig. 2. Model Overview: we encode the historical crowd flow into multiple graphs, and each graph is processed by a spatial block. The spatial block outputs are fused and then input into the temporal model, finally, the prediction result is obtained through a fully connected neural network.

fact, we can only consider the area we are interested in, such as all subway stations in the city, because this paper does not consider the existing road network, but uses the Euclidean distance and the Pearson correlation coefficients between regions to construct the graphs. Let $x_{\mathrm{n}}^{t} \in \mathbb{R}^{F}$ represent the crowd flow of region n at time interval t, where F denotes the number of input features. In this paper, F is equal to 2, which represents in-flow and out-flow [1], respectively. $\mathrm{X}_{\mathrm{t},:,:} = [x_1^t, x_2^t, \ldots, x_N^t] \in \mathbb{R}^{N \times F}$ presents the crowd flow of all regions at time interval t, Let $\mathrm{M}(\cdot)$ denote our model ,and crowd flow forecasting problem is defined as: given historical crowd flow for p time intervals, forecast the crowd flow of all regions at next time interval:

$$X^{pre} = M\left(X_{t-p:t-1,:,:}\right) \in \mathbb{R}^{N \times F}. \tag{1}$$

where X^{pre} denotes the forecasting results.

B. Graph Neural Networks

Graph neural network (GNN) was first proposed by Scarselli et al. [18], and many variants have been generated since then. GNNs can be divided into four classes: convolution, attention, gate, and skip connection according to the propagation steps [19]. The ChebNet proposed by Defferrard et al. [20] belongs to class convolution. For graph $G(V, E)$, V is the set of vertices of G, E is the set of edges of G, let $\mathrm{L} = \mathrm{D} - \mathrm{A}$ denote the Laplacian Matrix of G, D represents the degree matrix and A denotes the adjacent matrix, ChebNet can be expressed as:

$$\text{ChebNet}\ (\mathrm{x}, \mathrm{A}) \approx \sum_{\mathrm{k}=0}^{\mathrm{K}} \Theta_{\mathrm{k}} \mathrm{T}_{\mathrm{k}}(\tilde{\mathrm{L}})\mathrm{x} \tag{2}$$

Where $\tilde{\mathrm{L}} = \frac{2}{\lambda_{\max}}\mathrm{L} - \mathrm{I_N}$ denotes the scaled Laplacian matrix, $\lambda_{\max}$ denotes the maximum degree of G, $\mathrm{T}(\tilde{\mathrm{L}})$ is a K-order Chebyshev polynomial approximation to $\tilde{\mathbf{L}}$, $\Theta \in \mathbb{R}^{K}$ is a vector of Chebyshev polynomial coefficients and $\mathrm{ChebNet}(\cdot)$ denotes a ChebNet layer. Veličković et al. [9] propose graph attention networks (GAT), let $\mathrm{N(i)}$ denote the neighbor of vertex i, and GAT can be expressed as:

$$\mathrm{x}_{\mathrm{i}}' = \alpha_{i,i} W x_i + \sum_{j \in N(i)} \alpha_{i,j} W x_j \tag{3}$$

$$\mathrm{GAT}(\mathrm{x}, \mathrm{A}) = [\mathrm{x}_1', \mathrm{x}_2', \ldots, \mathrm{x}_{\mathrm{N}}'] \tag{4}$$

where $W \in \mathbb{R}^{F_{in} \times F_{out}}$ is a learnable parameter matrix, $\mathrm{GAT}(\cdot))$ denotes a GAT layer, and $\alpha_{i,j}$ is computed by:

$$\alpha_{i,j} = \frac{\exp\left(\mathrm{LeakyReLU}\left(a^T \left[Wx_i \| Wx_j\right]\right)\right)}{\sum_{k \in N(i)} \exp\left(\mathrm{LeakyReLU}\left(a^T \left[Wx_i \| Wx_k\right]\right)\right)} \tag{5}$$

where $a \in \mathbb{R}^{2F_{out}}$ is a learnable parameter vector.

C. Model Overview

As it is shown in figure 2, our model contains a spatial model and a temporal model to extract the spatial and temporal dependency of the crowd flow in turn. The spatial model includes a graph construction block, a spatial block composed

of a GAT and a ChebNet, and a multi-graph fusion block. It is worth noting that although only two types of graphs are used in figure 2, if there is other auxiliary information in actual application, e.g. urban subway networks, we can also construct corresponding graphs and add them to the graph construction block. The temporal model includes a shared weight Long Short-Term Memory(LSTM) and self-attention mechanism. We apply two different attention mechanisms, GAT and self-attention mechanism, to make the model pay more attention to important vertices and time intervals.

D. Spatial Model

1) Multi-Graph Construction Block: as shown in figure 1(b), the crowd flow of a region is affected by the crowd flow of adjacent regions and regions with high similarity in regional functions (such as two adjacent bridges). The distances were measured using Euclidean distance and Pearson correlation coefficient, respectively. We encode crowd flow into two K-NN unweighted directed graphs without auxiliary information. Obviously, compared to undirected graphs, directed graphs carry richer information. However, although weighted graphs carry richer information than unweighted graphs, we chose unweighted graphs because we believe that the weights of weighted graphs are not flexible enough, and with a GAT, the model can dynamically assign weight to neighboring vertices. The reason why we construct K-NN graphs instead of using a distance threshold is that the parameters of the K-NN graph are easier to estimate and adjust. Let $\mathrm{G} = (V, E)$, $|V| = N$, each vertex in the graph represents a region of the city, $A \in \mathbb{R}^{N \times N}$ denotes the adjacent matrix of G,and A is defined as :

$$A_{i,j} = \begin{cases} 1, \text{ if } \operatorname{rank}\left(\operatorname{dis}\left(v_i, v_j\right)\right) < k \\ 0, \text{ otherwise} \end{cases} \tag{6}$$

where $\operatorname{dis}\left(v_i, v_j\right)$ represents the distance between v_i and v_j, $\operatorname{rank}\left(\operatorname{dis}\left(v_i, v_j\right)\right)$ denotes the rank of $\operatorname{dis}\left(v_i, v_j\right)$ in descending order of S_i defined as equation (7):

$$S_i = \left\{\operatorname{dis}\left(v_i, v_j\right) | j = 1, \ldots, N\right\} \tag{7}$$

and k is the parameter of K-NN, in this paper, the distance is measured by the Euclidean distance between the center of the regions and the Pearson correlation coefficients computed by historical data, respectively.

$$\operatorname{dis}_{\mathrm{eu}}\left(v_i, v_j\right) = \begin{cases} \dfrac{1}{\text{Euclidean}\left(v_i, v_j\right)}, \text{ if } i \neq j \\ +\infty, \text{ if } i = j \end{cases} \tag{8}$$

$$\operatorname{dis}_{\mathrm{pe}}\left(v_i, v_j\right) = \frac{1}{F} \sum_{f=1}^{F} \operatorname{Pearson}\left(X_{:,i,f}, X_{:,j,f}\right) \tag{9}$$

where $\operatorname{Euclidean}(\cdot)$ denotes the Euclidean distance calculation function, and $\operatorname{Pearson}(\cdot)$ represents the Pearson correlation coefficient function. Through the above method, we construct two K-NN graphs with different adjacent matrices: A^{eu} and A^{pe}.

2) Spatial Block And Multi-Graph Fusion Block: after the graph construction block encodes the crowd flow into multiple K-NN unweighted directed graphs, these graphs are processed by spatial block separately, which include GAT and a ChebNet. GAT assigns weights to the neighboring vertices of each vertex, making the model pay more attention to important vertices, and then a ChebNet is employed to extract spatial dependency.We use different spatial blocks for different graphs to extract their spatial dependency, since the differences between each graph only exist in its edge set, the graphs processed by the spatial blocks are fused by uniting their edge sets. Let $\mathrm{M_{sp}}$ represent the spatial block,$\mathrm{M_{sp}}$ is defined as equation (10) and the spatial representation at time interval $t-q$: $X^{fu}_{t-q,:,:}$ can be expressed as equation (13):

$$\begin{aligned} M_{sp}\left(X_{t-q,:,:}, A\right) = & \operatorname{ReLU}(\operatorname{ChebNet}(\operatorname{ReLU} \\ & \left.\left(\operatorname{GAT}\left(X_{t-q,:,:}, A\right)\right), A\right)) \end{aligned} \tag{10}$$

$$X^{eu}_{t-q,:,:} = M^{eu}_{sp}\left(\left(X_{t-q,:,:}\right), \mathrm{A}^{\mathrm{eu}}\right) \text{ for } q = 1, \cdots, p \tag{11}$$

$$X^{pe}_{t-q,:,:} = M^{pe}_{sp}\left(\left(X_{t-q,:,:}\right), \mathrm{A}^{\mathrm{pe}}\right) \text{ for } q = 1, \cdots, p \tag{12}$$

$$\begin{aligned} X^{fu}_{q,:,:} = & \text{ ChebNet}\left(w_1 X^{eu}_{t-q,:,:} + w_2 X^{pe}_{t-q,:,:}, \right. \\ & \left.\left(A^{eu} \oplus A^{pe}\right)\right) \text{ for } q = 1, \ldots, p \end{aligned} \tag{13}$$

where $\mathrm{X}^{\mathrm{eu}} \in \mathbb{R}^{p \times N \times F'}$ and $\mathrm{X}^{\mathrm{pe}} \in \mathbb{R}^{p \times N \times F'}$are computed by different adjacent matrices, F' denotes the number of spatial block output channels, $\mathrm{X}^{\mathrm{fu}} \in \mathbb{R}^{p \times N \times F_{sp}}$, F_{Sp} denotes the number of fusion block output channels in equation (13), $w_1 \in \mathbb{R}$ and $w_2 \in \mathbb{R}$ are learnable parameters, $\oplus$ represents the element-wise logical OR operator.

E. Temporal Model

The temporal model includes a LSTM and a self-attention mechanism. We employ a shared weight LSTM among regions to model the temporal correlation, for weight sharing reduces the number of model parameters and saves training time. And then a self-attention mechanism is used to adaptively estimate the importance of the temporal representation of different time intervals:

$$H_{:,n,:} = LSTM\left(X^{fu}_{:,n,:}\right) \in \mathbb{R}^{p \times F_{tem}} \text{ for } n = 1, \cdots, N \tag{14}$$

$$a_{n,:} = \operatorname{softmax}\left(W_{a1} \tanh\left(W_{a2} H^{T}_{:,n,:}\right)\right) \text{ for } n = 1, \ldots, N \tag{15}$$

where $H_{:,n,:}$ denotes the n-th region's hidden states of LSTM, F_{tem} represents the number of spatial model output channels, $a_{n,:} \in \mathbb{R}^{\mathrm{p}}$ is the importance estimate of hidden states of different time intervals, $\mathrm{W_{a1}} \in \mathbb{R}^{p}$ and $\mathrm{W_{a2}} \in \mathbb{R}^{p \times F_{\mathrm{sp}}}$ are learnable parameters, the temporal model output $X^{\mathrm{tem}} \in \mathbb{R}^{N \times F_{\mathrm{tem}}}$ is calculated as:

$$X^{\mathrm{tem}}_{n,:} = \sum_{i=0}^{p} a_{n,i} H_{i,n,:} \text{ for } n = 1, \ldots, N \tag{16}$$

In the end, a fully connected layer is applied following the temporal model:

$$\mathrm{X}^{\mathrm{pre}} = \mathrm{X}^{\mathrm{tem}} W_{fc} \tag{17}$$

where $\mathrm{W_{fc}} \in \mathbb{R}^{\mathrm{F}_{tem} \times \mathrm{F}}$ is the parameter of the fully connected layer and $\mathrm{X}^{\mathrm{pre}} \in \mathbb{R}^{N \times F}$ denotes the forecasting result.

TABLE I
DETAILS OF DATASET TAXICQ AND BIKENYC

	TaxiCQ	BikeNYC
date span	03.01.2019∼06.30.2019	04.01.2014∼09.30.2014
invalid regions	141/512	47/128
interval(minute)	30	60
grid size	(16, 32)	(16, 8)

IV. EXPERIMENT

A. Datasets

Two real-world datasets were used to evaluate our model as follows:

- TaxiCQ: this is a self-collected dataset of Chongqing taxis for four months from March 1 to June 30, 2019. It records the movement of taxis in the main urban area of Chongqing by vehicle GPS device. This dataset is time-sensitive and can show the changes of crowd flow in real life and we follow research [1] to define the out-flow and in-flow. It contains 512 regions with a time interval of 30 minutes, each day is divided into 48 time intervals. Due to the complex terrain of Chongqing (including rivers and hills), 141 regions are invalid regions. The data of these regions stays zero at any time, but in order to apply this dataset on CNN-based models, we retain these invalid data. The grid size of TaxiCQ is (16, 32). We use the last ten days' data for testing and the rest for training.
- BikeNYC [21]: this dataset includes bike rental and return data of New York shared bike system from April 1 to September 30, 2014, the time interval is 1 hour, each day includes 24 time intervals, and the grid size is (16, 8), including 47 invalid regions. Similarly, we use the data of last ten days as for testing.

In table 1, we show the details of these two datasets.

B. Compared Models

The following models are employed to compare with ours:

- Historical average(HA): Historical average calculates the average value of historical data. e.g., the in-flow prediction at 10th interval on next Monday is the average of all in-flow at 10th interval of historical Mondays.
- ARIMA: Auto-Regressive Integrated Moving Average (ARIMA) is a time series prediction method.
- ConvLSTM [22]: ConvLSTM replaces all the internal operations of LSTM with convolution operations, we deployed two layers of ConvLSTM and a fully connected layer to get forecasting results.
- ST-ResNet [1]: A CNN-based model using residual neural network to capture trend, periodicity, and closeness information
- E3D-LSTM [4]: E3D-LSTM integrates LSTM and 3D convolution, strengthens the long-term memory ability of LSTM and fuses self-attention mechanism. It achieves good performance in the field of spatio-temporal modeling.

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON BIKENYC AND TAXICQ.

Method	BikeNYC		TaxiCQ	
	MAE	RMSE	MAE	RMSE
HA	2.81	8.17	12.27	25.89
ARIMA	3.57	11.08	10.11	19.82
ConvLSTM	3.16	7.29	6.53	13.93
ST-ResNet	2.56	6.33	6.84	13.96
E3D-LSTM	3.02	6.99	7.08	14.20
DCRNN	2.46	6.30	6.61	13.87
STGCN	2.51	6.31	6.89	14.18
MGNN (ours)	**2.27**	**5.95**	**6.07**	**12.73**

- DCRNN [7]: DCRNN is a graph convolution based model with graph constructed with road networks, and the graph convolution is embedded in the encoder-decoder architecture to capture spatio-temporal dependency.
- STGCN [6]: STGCN is a spatio-temporal graph convolution based model for traffic forecasting.

C. Experiment Results

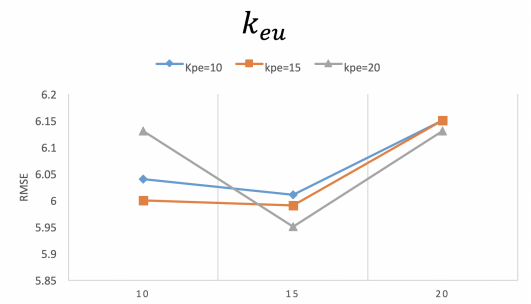
Table2 demonstrates the result of our work(MGNN) and compared models on TaxiCQ and BikeNYC. We can observe that the deep learning approaches are significantly better than the statistical methods because the former can effectively extract complex spatio-temporal dependency. On the other hand, graph neural network-based models (DCRNN, STGCN, MGNN) outperform models based on CNN (ST-ResNet, E3D-LSTM, ConvLSTM) in most evaluation metrics, indicating that graph neural networks have a better ability to capture spatial dependency. The model we proposed achieved the best results in both MAE and RMSE. More specifically, our work gains 7.72%, 8.17% relative improvement in MAE, and gains 5.56%, 8.22% relative improvement in RMSE. The results indicate that our model outperforms state-of-the-art in crowd flow forecasting.

D. Influence Of K-NN Parameters

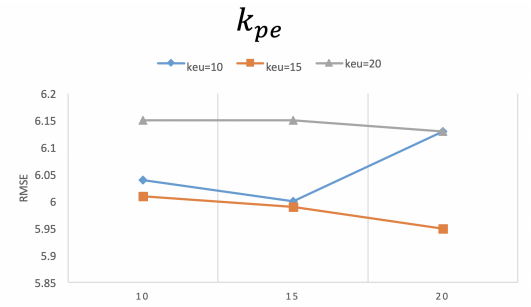
Two K-NN graphs are constructed in the spatial model with different parameters $\mathrm{k_{eu}}$ and $\mathrm{k_{pe}}$. In figure 3, we show the effect of different K-NN parameters on the forecasting results. We find that the more edges don't mean better performance and the predictions are more sensitive to changes in $\mathrm{k_{eu}}$, when $\mathrm{k_{pe}}$ stays the same, different $\mathrm{k_{eu}}$ can cause an average relative gap of 2.6% in RMSE, while the ratio for $\mathrm{k_{pe}}$ is 1.2%, this is also the reason why the CNN-based models can achieve good results although they only capture the spatial dependency of neighboring regions. The best performance is obtained when $\mathrm{k_{eu}}$ is equal to 15 and $\mathrm{k_{pe}}$ is equal to 20.

E. Effect of Attention Mechanism

We employ GAT and self-attention mechanism to extract dynamic spatio-temporal correlations in our model and we get 3 variants by removing these two attention mechanisms: MGNN-G (with GAT only), MGNN-S (with self-attention only) and MGNN-E (without any attention mechanisms). In



(a)



(b)

Fig. 3. Influence of K-NN parameters on BikeNYC in RMSE

TABLE III
PERFORMANCE OF MGNN AND ITS THREE VARIANTS ON TAXICQ IN RMSE

	RMSE
MGNN-E	13.47
MGNN-S	13.24
MGNN-G	12.94
MGNN	**12.73**

Table 3, we show the forecasting results of MGNN and its three variants on the dataset TaxiCQ. Obviously, it can be seen that both the GAT at the spatial level and the self-attention mechanism at the temporal level make the prediction result more accurate. More specifically, compared with MGNN-E, MGNN-S, MGNN-G, and MGNN increase by 1.7%, 3.9%, and 5.5% relative improvement in RMSE, respectively. It indicates that attention mechanisms do make the model pay more attention to important vertices and time intervals.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel graph neural network-based crowd prediction model which fused spatial and temporal attention mechanisms. Experiments on two large scale real-world datasets indicate that our model outperforms other state-of-the-art models. Moreover, we intend to introduce heterogeneous graph neural networks to our model. In heterogeneous graph neural networks, nodes will be divided into several classes, and heterogeneous graphs will contain richer information. In addition, we also consider applying this model to other spatiotemporal modeling fields such as social networks and environmental monitoring.

VI. ACKNOWLEDGEMENT

This research is supported by National Natural Science Foundation of China (41571401) and Chongqing Natural Science Foundation (cstc2014kjrcqnrc40002).

REFERENCES

[1] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artificial Intelligence*, vol. 259, pp. 147–166, 2018.

[2] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, and Z. Li, "Modeling spatial-temporal dynamics for traffic prediction," *arXiv preprint arXiv:1803.01254*, 2018.

[4] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3d LSTM: A model for video prediction and beyond," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=B1lKS2AqtX

[5] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 922–929.

[6] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 2018, pp. 3634–3640. [Online]. Available: https://doi.org/10.24963/ijcai.2018/505

[7] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=SJiHXGWAZ

[8] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *2019 AAAI Conference on Artificial Intelligence (AAAI'19)*, 2019.

[9] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[10] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[11] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[12] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through kalman filtering theory," *Transportation Research Part B: Methodological*, vol. 18, no. 1, pp. 1–11, 1984.

[13] C. M. Kuchipudi and S. I. Chien, "Development of a hybrid model for dynamic travel-time prediction," *Transportation Research Record*, vol. 1855, no. 1, pp. 22–31, 2003.

[14] H. Sun, H. X. Liu, H. Xiao, R. R. He, and B. Ran, "Use of local linear regression model for short-term traffic forecasting," *Transportation Research Record*, vol. 1836, no. 1, pp. 143–150, 2003.

[15] C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 276–281, 2004.

[16] B. Shen, X. Liang, Y. Ouyang, M. Liu, W. Zheng, and K. M. Carley, "Stepdeep: a novel spatial-temporal mobility event prediction framework based on deep neural network," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 724–733.

[17] Z. Yuan, X. Zhou, and T. Yang, "Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 984–992.

[18] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[19] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[20] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[21] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "Dnn-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2016, p. 92.

[22] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.