Passenger Demand Prediction With Cellular Footprints

Jing Chu, *Student Member, IEEE*, Xu Wang, *Student Member, IEEE*, Kun Qian, *Student Member, IEEE*, Lina Yao, *Member, IEEE*, Fu Xiao, *Member, IEEE*, Jianbo Li, and Zheng Yang, *Senior Member, IEEE*

Abstract—Accurate forecast of citywide passenger demand helps online car-hailing service providers to better schedule driver supplies. Previous research either uses only passenger order history and fails to capture the deep dependency of passenger demand, or is restricted on grid region partition that loses physical context. Recent advance in mobile traffic analysis has fostered understanding of city functions. In this article, we propose FlowFlexDP, a demand prediction model that integrates regional crowd flow and applies to flexible region partition. Analysis on a cellular dataset covering 1.5 million users in a major city in China reveals strong correlation between passenger demand and crowd flow. FlowFlexDP extracts both order history and crowd flow from cellular data, and adopts Graph Convolutional Neural Network to adapt prediction for regions of arbitrary shapes and sizes in a city. Evaluation on a large scale data set of 6 online car-hailing applications from cellular data shows that FlowFlexDP accurately predicts passenger demand and outperforms the state-of-the-art demand prediction methods.

Index Terms—Machine learning, prediction methods, predictive models, mobile computing, communication systems, mobile communication

1 Introduction

During the past decades, people have increasingly migrated from rural into urban areas following the rapid pace of worldwide urbanization. While urbanization has some advantages such as advancing technologies, employment opportunities, infrastructures and economic growths, it also results in the problem of how to balance the limited public resources and rapidly increasing populations. 'Smart City' is a solution that utilizes different types of data to help manage resources intelligently and efficiently and in this paper we focus on traffic and transportation sector, which has been transformed by the evolution of the Internet and online car-hailing service. Online car-hailing service, for its efficiency and comfort, performs as pivotal public transportation in the mobile age [1]. To hire a vehicle, a passenger can simply submit his desired pick up location and the destination to the online service provider (e.g., Uber, DiDi, Lyft), who then delivers the request to close-by drivers. Despite its convenience, the online car-hailing service still suffers from the imbalance between demand of passengers and supply of drivers in some local regions. For example, a driver hardly gets any request during the empty cruise, as few passengers near his cruise route request the service. Meanwhile, a passenger finds it is difficult to get the ride, especially in bad weather and rush hours, as overwhelmingly large number of requests are sent at the same time in the same region. Hence, it is necessary yet challenging for service providers to predict demand of passengers, in order to foreseeingly schedule drivers.

Existing works have promoted passenger demand prediction with elaborate models, including parametric models [2], [3] and neural network based models [4], [5]. The main idea is to relate passenger demand in different spatial region and time interval with local historical passenger order data [6], supplemented with weather [5] and traffic [7] data. However, the major limitation of using passenger order data lies in its scale; that is, the passenger coverage of the order data is restricted within the group of passengers who install the specific car-hailing apps on their mobile devices. Furthermore, the order data cannot fully characterize the passenger demand in local regions, which is potentially impacted by characteristics of region populations (e.g., crowd flow, human mobility and behaviour).

As mobile phones deeply penetrate everyday life of human, cellular data accommodates a larger user population and acts as an ideal tool for analysis of cities operation, such as cellular traffic prediction [8], city function discovery [9] and population composition [10]. This paper seeks to advance the state-of-the-art of passenger demand prediction by modelling with cellular data. Note that cellular data not only reveals historical information of passenger order, which can be resolved from the URL of cellular packets, but also characterizes the region populations from various aspects. By integrating multi-dimensional features available in cellular data, passenger demand can be more accurately predicted. However, passenger demand still has some complex properties, and leads to challenges for prediction:

- Spatial and temporal correlation. The passenger demand in specific region is correlated with not only historical demand in this region, but also that in other

- *Jing Chu, Xu Wang, Kun Qian, and Zheng Yang are with the School of Software Engineering, Tsinghua Universtiy, Bejing 100084, China. E-mail: j-zhu16@mails.tsinghua.edu.cn, xu-wang15@mails.thu.edu.cn, {qiank10, hmilyyz}@gmail.com.*
- *Lina Yao is with the School of Computer Science and Engineering, UNSW, Kensington, NSW 2052, Australia. E-mail: theresa0125@gmail.com.*
- *Fu Xiao is with the College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China. E-mail: xiaof@njupt.edu.cn.*
- *Jianbo Li is with the Computer Science and Technology College, Qingdao University, Qingdao 266071, China. E-mail: lijianbo@188.com.*

Manuscript received 14 Apr. 2019; revised 20 Apr. 2020; accepted 18 June 2020. Date of publication 26 June 2020; date of current version 3 Dec. 2021. (Corresponding author: Zheng Yang.)
Digital Object Identifier no. 10.1109/TMC.2020.3005240

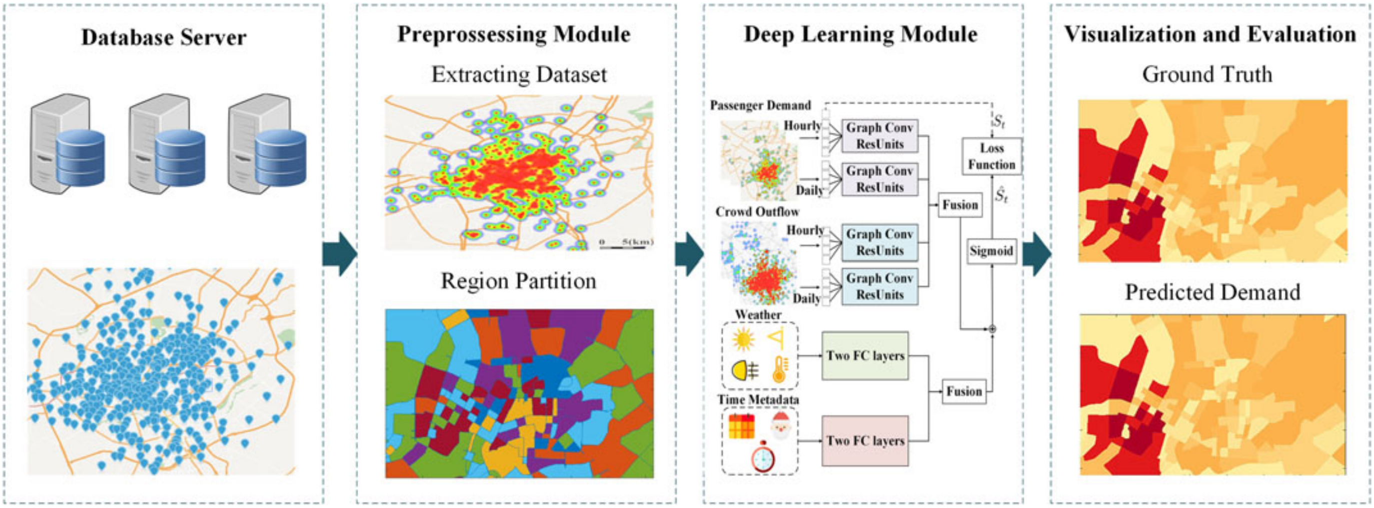


Fig. 1. System architecture. The system consists of 4 modules: database module, pre-processing module, deep learning module, and visualization and evaluation module.

regions, whose contributions are inversely proportional to the distances between two regions [11]. Meanwhile, the impact of the historical demand is also inversely proportional to the length between two time points. Furthermore, the passenger demand exhibits periodicity, with rush hours in the morning and the evening, and trough in the midnight.

- Region characteristics. Besides the historical demand, the passenger demand in a specific area is also dependent on the characteristics in this area, such as weather [12], traffic and crowd flow. Using only historical order and weather data, existing works [6] cannot predict and further adapt to potential huge passenger demand due to burst crowd outflow (e.g., matches, concerts). Hence, in addition to historical order, the crowd flow should be considered.
- Region partition. Existing works [5] partition a city in to regular grids (i.e., regions), and apply CNN to the regions to achieve prediction. Yet in practice, the city is partitioned into geographically irregular regions by the government; and drivers are scheduled to cruise in predefined functional regions. Thus, it is necessary to predict passenger demand on a flexible region partition of the city.

To address these challenges, we propose FlowFlexDP, a deep learning model that predicts passenger demand with cellular data. First, FlowFlexDP divides the demand series into half-hourly correlation series and daily correlation series to model temporal correlation, and uses distances between regions to model spatial correlation. Second, FlowFlexDP extracts historical order records, as well as estimates crowd flows of regions from cellular data. Meanwhile, factors including weather, holidays, time-of-day, and day-of-week are also carefully considered in FlowFlexDP. Last but not least, FlowFlexDP adopts Graph Convolutional Neural Networks (GCNN) [13] to adapt to flexible region partition, and uses residual network [14] to deepen the neural network, in order to learn the relations between far away regions.

In summary, the main contributions are as follows:

- We demonstrate cellular data as a rich data source for passenger demand prediction, which have been largely overlooked and unexplored previously. We extract historical order, crowd flow from cellular data, and develop the prediction model accordingly. As far as we are aware of, it is the first work that uses crowd flow information from cellular traffic for passenger demand prediction.
- We devise a novel neural network architecture that uses GCNN to enable demand prediction on flexible region partition, which is more realistic than regular partition.
- We extract a large scale real passenger demand data set of 6 online car-hailing service from cellular traffic data, and conduct extensive experiments. The results show that FlowFlexDP achieves a more accurate prediction accuracy. By considering crowd flow data, and accommodating flexible region partition, FlowFlexDP outperforms existing previous models.

2 System Overview and Data Source

In this section, we briefly introduce the system architecture and our data source.

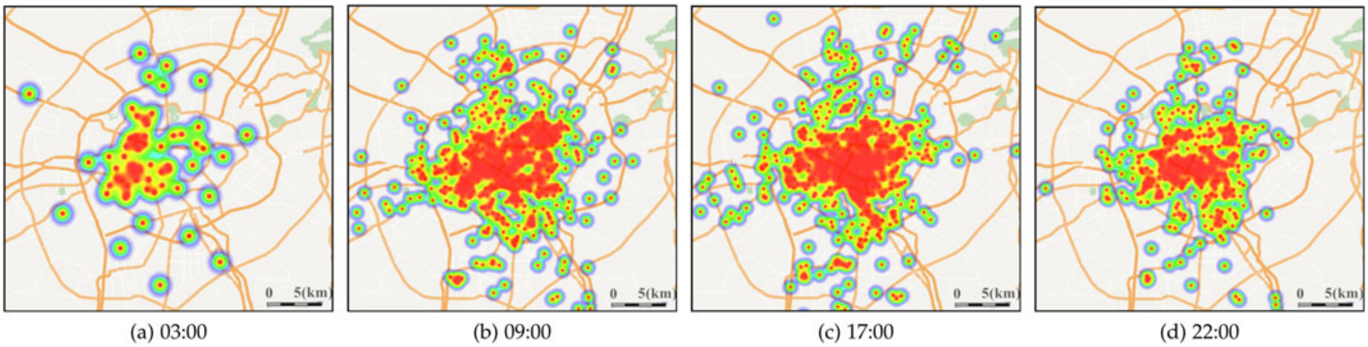
2.1 System Overview

As shown in Fig. 1, our prediction system consists of 4 modules: database module, pre-processing module, deep learning module, and visualization and evaluation module. The database server module stores the big cellular data and provides retrieval and aggregation services for fast pre-processing. In the pre-processing module we partition the city, clean the data and extract the passenger information and crowd flow information. The deep learning module FlowFlexDP is the key component of our system and it takes four kinds of data to get precise passenger demand prediction. The detail of FlowFlexDP will be introduced in later section. In the visualization and evaluation module, we evaluate the performance of the system and generate real-time heat maps of passenger demand to advice drivers where to pick up passengers.

2.2 Data Source

2.2.1 Cellular Data

Our dataset was collected by a major cellular carrier in a big city of China. Fig. 3 shows the architecture of a typical



(a) 03:00 (b) 09:00 (c) 17:00 (d) 22:00

Fig. 2. Spatial distribution of urban passenger demand at different times during a day.

cellular network. To facilitate network management, a monitoring system is deployed for operators to analyze data traffic, monitor network performance and detect anomalies. The monitoring system consists of a detector and a central, which records important information for network diagnosis and forensics. Each record in the dataset contains bidirectional flow information with the following key fields: a unique anonymized user ID, the flow create time, the flow connected cell tower ID, App ID, device ID, uplink traffic, downlink traffic, URL, URI, etc. The dataset covers from Dec 5th, 2016 to Feb 4th, 2017 and contains more than 2.9×10^{10} mobile cellular records with 8×10^3 cell towers and 1.5×10^6 mobile users covered. The wide coverage in mobile users and cell towers promises to capture comprehensive spatial-temporal dynamics in passenger demand and crowd flows.To the best of our knowledge, the dataset is one of the largest urban-scale cellular traffic dataset in terms of the number of mobile users and cell towers, which requires a high-performance server cluster for efficient analysis and prediction. In our platform, we take Greenplum as our dataset platform with three high-performance data servers.

Since our aim is to predict passenger demand, we use the location of the cell tower as an estimate for each mobile user. The cell tower ID in each record corresponds to a cell tower geographical coordinates. Note that a mobile device may not be associated to the nearest cell tower [15]. However, a location accuracy of 70m is achievable with the coverage information of cell towers [16].

In order to understand the meaning behind the URLs in our dataset, we did a field experiment to capture the HTTP and SSL/HTTPS header by some HTTP proxy tool (*i.e.*, Charles [17]) when taking online car-hailing applications. We did the experiments with 6 applications, including DiDi taxis and other types of cars (e.g., DiDi Express, DiDi Premier), Uber, 51 Yong Che, Yi Dao Yong Che, UCAR and CAOCAO. The collected URLs are matched with the volunteer's user behaviour. The corresponding results of ordering a taxi in DiDi platform are shown in Table 1. The records with the same URL and URI are recognized as the citywide passenger demand, which cover more than 5.4×10^5 passengers. The spatial distribution of urban passenger demand at different times during a day is shown in Fig. 2, which depicts the changing of urban passenger demand of different times (03:00, 09:00, 15:00, 22:00) on a weekday. Specifically, the passenger demand in the city center is much heavier throughout the day. Overall the passenger demand before dawn is low as most residents in the city are sleeping at that time. At 09:00 the demand at most areas starts to increase as people need to go to work. Similarity, at 17:00 the demand increases again because people go off work. Surprisingly, we find the passenger demand is also high at midnight. The reason might lie in the fact that it is a metropolis and there can be rich night life. As for the crowd flow data, we get it with the method introduced in the problem formulation section.

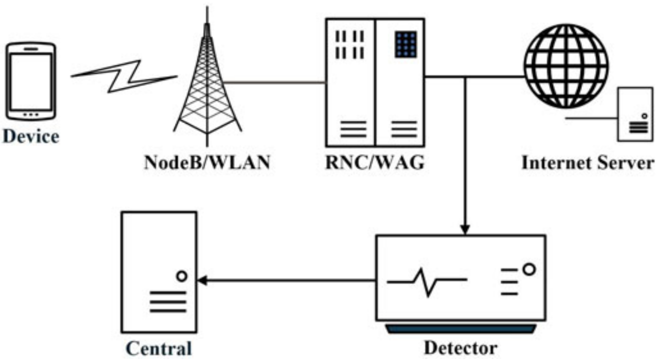


Fig. 3. An illustration of cellular network architecture and data monitoring system.

2.2.2 External Data

Besides cellular data, weather conditions (*i.e.*, weather state, temperature, wind speed and visibility), time-of-day, day-of-week and holidays are utilized for prediction due to their key influence on passenger demands. For better prediction, we collect the corresponding weather data from the Dark Sky API [18]. The overview of the weather data is illustrated in Table 2.

3 Problem Formulation

In this section, we introduce the terms of use and formalize the passenger demand prediction problem.

TABLE 1
Corresponding Results of Ordering Taxis

Action	URL(api.diditaxi.com.cn/)
Place an order	api/v2/p_neworder
Check order status	api/v2/p_getorderstatus
Get order detail	api/v2/ p_getorderdetail
Request payment info	api/v2/p_getpayinfo
Comment	api/v2/p_comment

TABLE 2
Weather Data

Weather data	Data condition
Weather states	11 types
Temperature (°C)	[-21.24, 6.11]
Wind speed (m/s)	[0, 7.15]
Visibility (km)	[0.16, 16.09]

Definition 1 (Passenger Demand). *Let $R = r_1, r_2, \ldots, r_N$ be the set of N region, and the passenger demand of a region r_i in a given time window $[t, t + \Delta t]$ is defined as $D_t(r_i)$, where $D_t(r_i)$ is the number of passenger taking a taxi or car of car-hailing platforms at region r_i. And $D_t = \{D_t(r_i)|i = 1, \ldots, N\}$ represents the sequence of passenger demand of all regions at time t.*

Definition 2 (Crowd Outflow). *Crowd Outflow [19] of a region means the number of people leaving the region in a time range. Similarly, we use $P_t(r_i)$ to denote the set of people located inside the region r_i at a given time interval $[t, t + \Delta t]$. In order to reduce the impact of passersby, we distinguish the users leaving after long time staying (e.g., leaving after work) from passersby by analyzing the time length the user stays in a region. The crowd outflow of the region r_i at the time t is defined as $C_t(r_i) = P_t(r_i) \setminus P_{t+1}(r_i)$. And $C_t = \{C_t(r_i)|i = 1, \ldots, N\}$ denotes the sequence of outflow of all regions at time t.*

Given the historical observations, including the number of passenger demand $D_1, D_2, \ldots, D_t$, crowd outflow $C_1, C_2, \ldots, C_t$ and the external data, our goal is to build a deep learning model to predict D_{t+1}.

4 Region Partition

Although many prediction approaches for spatial data require region partition in the form of grid, flexible region partition is more attractive since regions with semantic or administrative meanings often demonstrate irregular shapes rather than grids. As a result, flexible region partition is expected for city planners, ride service managers, last but not the least, car drivers. The existing region partition methods mainly including road network based partition [20], POI cluster based partition [21] and grid region partition, but these methods have certain limitations for the prediction of passenger demand. Using only the road network, the functional and human flow characteristics of the region cannot be well concerned. The POI cluster based partition method is affected by the sparseness of the POI, while the POI features and the flow characteristics are largely inconsistent. The grid form region partition method divides the city into square blocks, which does not have much specific meaning, only for realization of the model with convolutional neural network. To extract semantic regions, we design our region partition algorithm partitioning the whole city into regions by the combination of road network and crowd flow characteristics. Specifically, the region partition method including fine-grained urban region partition based on road network and fine-grained region clustering based on regional crowd outflow.

Initially, we partition the entire city into blocks by the road network. A road network refers to a road system composed of various roads interconnected and interwoven into a network in a certain area. The formation and development of urban road networks is related to the development history of politics, economy, production and life in the city. Therefore, the use of road networks allows for a more scientific division of urban areas. Particularly, we extract the road network from OpenStreetMap [22] and convert the map to a binary image where roads are represented by black lines. Further, we apply "bwlabel" algorithm [23] to detect the connected components as partition result. Fig. 4 shows the primary region partition and we get 412 regions.

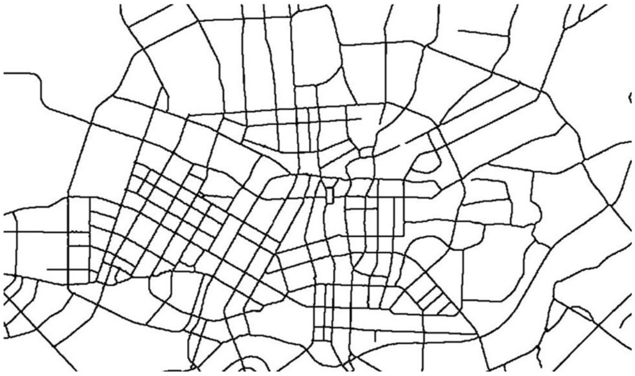


Fig. 4. Region partition with road network.

As can be seen from Fig. 4, partition with only road network is insufficient because the divided area is too small and the crowd outflow characteristics of the area are not taken into consideration. Accordingly, after road network fine-grained partition, the divided region is further clustered with the characteristics of the crowd outflow.

Regional crowd outflow can well reflect the functionality and attraction of the region. In order to cluster the regions, we first use the Minkowski distance to calculate the similarity between the sequences of crowd outflow of each region. The Minkowski distance is a metric in a normed vector space, which can be used to calculate the distance of the sequences. The Minkowski distance of two n-dimensional variables $r_i(C_{i1}, C_{i2}, \ldots, C_{in})$ and $r_j(C_{j1}, C_{j2}, \ldots, C_{jn})$ is defined as:

$$d_{ij} = \sqrt[p]{\sum_{k=1}^{n} |C_{ik} - C_{jk}|^p}, \tag{1}$$

where d_{ij} is the distance between crowd outflow sequence of region r_i and region r_j and $p = 2$ to calculate euclidean distance. Then we use the spectral clustering algorithm [24] to cluster the fine-grained areas to achieve the final urban region partition. Spectral clustering has emerged recently as a popular clustering method that uses eigenvectors of a matrix derived from the data. Spectral clustering makes use of the spectrum of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. Compared with the k-means clustering method [25], it has stronger adaptability to data distribution. At the same time, for region clustering, it is necessary to cluster the regions that are spatially adjacent and have similar crowd outflow characteristics. The spectral clustering method can better realize the clustering of similar regions. Formally, we model the city regions as a graph $G = (V, E, A)$ where $V = \{r_i|i = 1, 2, \ldots, n\}$ is a finite set of regions. E is the set of edges connecting regions, and A is an edge similarity matrix, which is

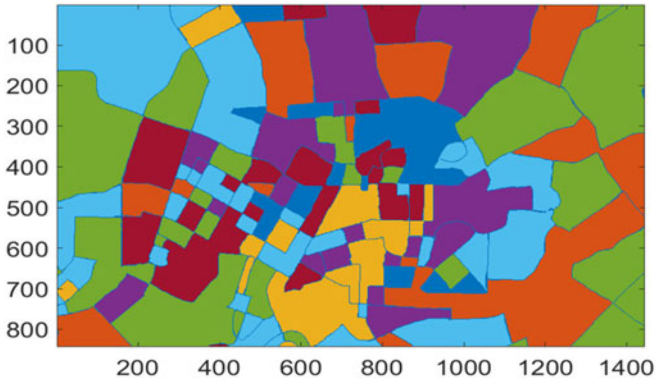


Fig. 5. Cluster region by crowd outflow.

nonnegative and symmetric. The Minkowski distance between the crowd outflow sequences of each region d can be used to calculate edge similarity matrix A, which is defined as:

$$A_{ij} = A_{ji} = \begin{cases} 0 & r_i \text{ and } r_j \text{ are not adjacent} \\ exp(-\frac{{d_{ij}}^2}{2\sigma^2})r_i & \text{and } r_j \text{ are adjacent} \end{cases}, \tag{2}$$

where d_{ij} is the Minkowski distance between the two sequences, σ is the standard deviation of all Minkowski distances. Then the graph Laplacian matrix is defined as:

$$L = D - A, \tag{3}$$

where D is the diagonal matrix$D_{ii} = \sum_j A_{ij}$ Then by using normalized cuts algorithm [26], the symmetric normalized Laplacian defined as:

$$L^{norm} = I - D^{-1/2}AD^{-1/2}. \tag{4}$$

A mathematically equivalent algorithm [27] takes the eigenvector corresponding to the largest eigenvalue of the random walk normalized adjacency matrix $P = D^{-1}A$. We extract the eigenvector f corresponding to the smallest k_1 eigenvalues, which further be clustered by K-Means clustering method. The final partition is shown as Fig. 5 and the number of regions is 75.

5 Deep Learning Architecture

In this paper, we design a novel deep learning architecture FlowFlexDP to model and predict the passenger demand for each region. Fig. 6 illustrates the architecture of FlowFlexDP.

In our model, in order to learn from the historical passenger demand and crowd outflow, historical passenger demand, historical crowd outflow and global external factors are fed as the input. Historical passenger demand is decomposed into passenger demand half-hourly sequence and passenger demand daily sequence. Similarly, historical crowd outflow is decomposed into crowd flow half-hourly sequence and crowd outflow daily sequence. Each of the sequence components is fed into deep neural networks, which are made up of graph convolutional neural networks and ResUnits. Global external factors (*i.e.*, weather data and time metadata) are fed into two-layer fully-connected neural networks respectively. Next, we apply the fusion operation to the output of the four sequence components, and merge the fusion output with the outputs of the fully-connected neural networks. The detail of our model is explained as below.

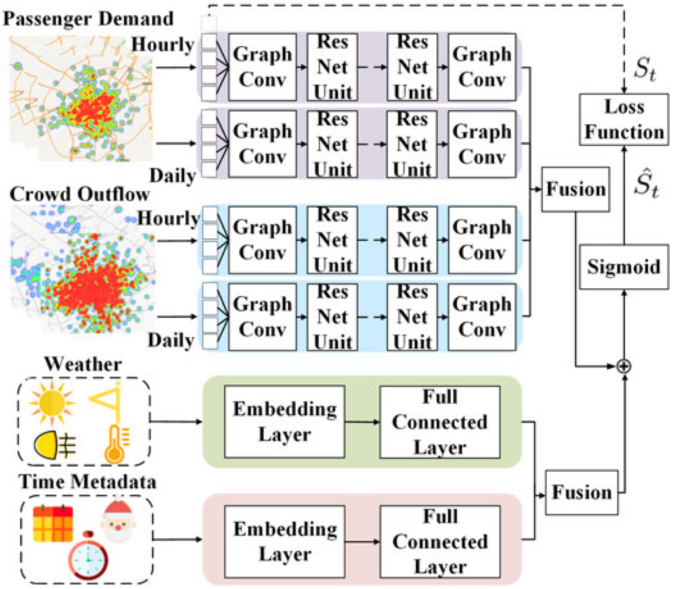


Fig. 6. The architecture of FlowFlexDP.

5.1 Graph Convolutional Neural Network

As we know, the passenger demand in one region is not only determined by the context of the region, but also influenced by the context in other regions [11]. And the dependency between nearby regions is stronger than distant regions. To model the spatial dependency, we apply graph convolutional neural network (GCNN) in our network architecture. Graph convolutional neural network is designed for generalizing convolutional neural networks from low-dimensional regular grids where image, video and speech are represented, to high-dimensional irregular domains to learn the local, stationary and compositional spatial-temporal features.

Formally, we model the city regions as a weighted graph $G = (V, E, W)$, where $V = \{r_i | i = 1, 2, \ldots, N\}$ is a finite set of regions. $E = \{(r_i, r_j) | r_i \text{ and } r_j \text{ are adjacent}\}$ is a set of edges. W is the adjacency matrix. The normalized graph Laplacian L of G is defined as:

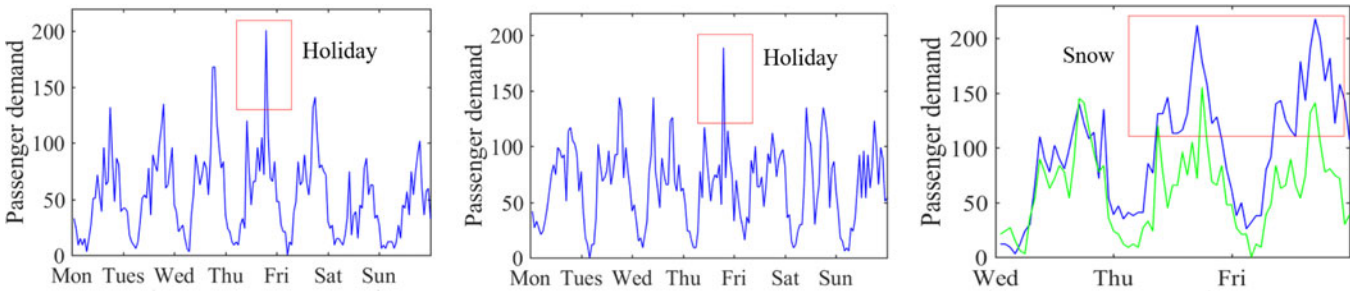
$$L = I_n - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}, \tag{5}$$

where D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$ and I_n is the identity matrix. Since L is a real symmetric positive semidefinite matrix, it has a complete set of orthonormal eigenvectors $\{u_l\}_{l=1}^n$ and real nonnegative eigenvalues $\{\lambda_l\}_{l=1}^n$. The filter operation g on a graph signal $x \in R^{n \times d_x}$ is $g_\theta(\Lambda) = diag(\theta)$, where $\theta \in R^n$ is a vector of Fourier coeffcients learned by the neural networks. The convolutional operation on x is defined as:

$$y = g_\theta(L)x = g_\theta(U\Lambda U^T)x = Ug_\theta(\Lambda)U^T x, \tag{6}$$

where $U = [u_1, \ldots, u_n] \in R^{n \times n}$ is the matrix of eigenvectors and $\Lambda \in R^{n \times n}$ is the diagonal matrix of eigenvalues of the normalized graph Laplacian L.

Furthermore, computing the eigen decomposition of L is prohibitively expensive for large graphs. To overcome this



(a) Office Region

(b) Residential Region

(c) Dec 21st-23th, 2016 (blue), Jan 4th-6th, 2017 (green)

Fig. 7. Passenger demand of one week in different regions and effects of holidays and weather: (a) passenger demand of one week in office region; (b) passenger demand of one week in residential region; (c) effect of snow in office region.

problem, a polynomial filter is used and defined as:

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k, \tag{7}$$

where the parameter $\theta \in R^K$ is a vector of polynomial coefficients and K is a super parameter. Since $d_G(i,j) > K$ implies $(L_K)_{i,j} = 0$, where d_G is the shortest path distance. Consequently, spectral filters represented by K-order polynomials of the Laplacian are exactly K-localized.

In our architecture, we apply GCNN with K-order polynomial filter to learn the spatial dependency. Since K-order polynomial filter can learn the spatial K-localized spatial dependency, more deeper stacked GCNN layers can learn the city-scale spatial dependency.

5.2 GCNN With Residual Learning (ResGCNN)

In order to capture the long-distance spatial dependency between regions, we need to use a lot of consecutive convolutional layers. Although training through the ReLU activation function and regularization can be more effectiveness, we still need to solve the problem of deeper networks. So we apply residual learning [14] to GCNN as traditional CNN-based residual learning. Residual learning allows deeper neural networks to be effectively trained and has made great success in deep learning area. A residual unit can be represented as:

$$x_{l+1} = x_l + F(x_l), \tag{8}$$

where x_{l+1} and x_l are the input and output of the lth unit in the network, F is the mapping function. Fig. 8 shows the structure of residual unit of GCNN. In our work, we add shortcut connections for each two-layer GCNN. The component works as follows: the input data will first be fed into a GCNN layer to be encoded into fixed dimension tensor and then the output will continually be fed into L residual GCNN layers. Finally, the output is input into another GCNN layer to encode the dimension for fusion operation.

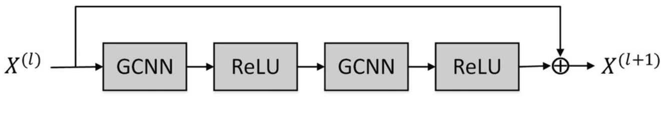


Fig. 8. The residual unit of GCNN.

5.3 Hourly and Daily Sequences Fusion

ResGCNN can capture the spatial dependencies, but it fails to capture the temporal dependencies. In order to learn the spatial-temporal dependencies, we divide the passenger demand sequences into two parts, the half-hourly sequence and daily sequence. For half-hourly sequence, there is no doubt that the passenger demand of a time is affected by the previous hours. As for daily sequence, Figs. 7a and 7b depict the time dynamics of passenger demand of office region and residential region in one week, respectively. Obviously that passenger demand has strong time dependency and periodicity for days. So by extracting the daily sequence we can learn this periodicity.

Specifically, we aggregate the passenger demand by half hour time interval. The corresponding half-hourly sequence at t is $[D_{t-h}, D_{t-(h-1)}, \ldots, D_{t-1}]$ and daily sequence at t is $[D_{t-T\cdot d}, D_{t-T\cdot(d-1)}, \ldots, D_{t-T}]$, where h is the length of half-hourly time intervals and d is the length of daily sequence, T=48 for twenty-four hours. With the GCNN layer and and L residual GCNN layers, the output of hourly sequence is D_{ho} and for daily sequence is D_{do}.

Regions usually show different time characteristic from other regions. As shown in Figs. 7a and 7b, in office region, the evening peak values are much higher than morning peak values while the peak values on weekdays are much higher than weekends, which makes sense for people do not need to work on weekends. In residential region, the peak values on weekends are similar with the ones on weekdays. So the half-hourly and daily sequences may have different influence degrees. We fuse the two components with parametric-matrix-based fusion [28], which is defined as

$$D_o = W_{ho} \circ D_{ho} + W_{do} \circ D_{do}, \tag{9}$$

where $\circ$ is Hadamard product, which means element-wise multiplication. W_{ho} and W_{do} are learnable parameters, which can change the influence degrees of half-hourly and daily sequences.

5.4 Crowd Outflow Sequences Fusion

Considering if the crowd outflow of a region increased dramatically, such as after a concert the crowd outflow of a region can be far more than the other time of the region, resulting in a substantial passenger demand increase. But prediction only with the historical passenger demand sequence sometimes can not perceive the demand change

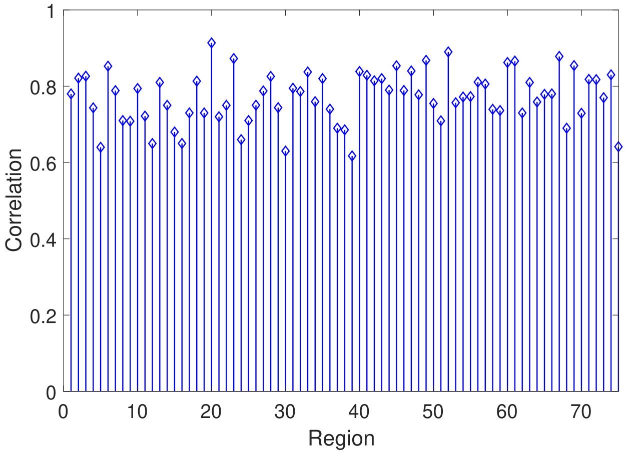


Fig. 9. The cross correlation coefficient of passenger demand sequence and crowd outflow sequence in each region.

information caused by crowd outflow. So we consider combining the regional crowd outflow data to achieve better passenger demand prediction. As far as we know, this is the first work which adding the crowd outflow data to predict passenger demand. Fig. 9 illustrates the strong correlation between crowd outflow and passenger demand. Since passengers are a small part of crowd outflow, the outflow information is very helpful for passenger demand prediction. It means that besides demand history, crowd flow may act as a new data source for demand prediction. Similarly, we define the half-hourly sequence and daily sequence of crowd outflow at time t as $[C_{t-h}, C_{t-(h-1)}, \ldots, C_{t-1}]$ and $[C_{t-T\cdot d}, C_{t-T\cdot(d-1)}, \ldots, C_{t-T}]$, where h is the length of half-hourly time intervals and d is the length of daily sequence, T=48 for twenty-four hours.

For these two kinds of sequences, we feed them into another two deep neural networks, respectively. The output of crowd outflow half-hourly sequence is C_{ho} and for daily sequence is C_{do}. These sequences are fused with parametric-matrix-based fusion [28]. This fusion method uses Hadamard product to element-wise multiply learnable parameter and output vector, which is defined as

$$C_o = W_{hc} \circ C_{ho} + W_{dc} \circ C_{do}, \tag{10}$$

where $\circ$ is Hadamard product, W_{hc} and W_{dc} are learnable parameters. And we fuse the output of passenger demand sequences and crowd outflow sequences as

$$O_{dc} = C_o + D_o, \tag{11}$$

where O_{dc} is the output of the fusion method.

5.5 External Factors Fusion

Fig. 7a shows the influence of daytype (*i.e.*, weekday or weekend) for passenger demand, Fig. 7c illustrates that in comparison with sunny days, the demand on snowy days increase sharply, and Fig. 7b demonstrates the influence of holiday. These observations insight us to use external data for better prediction accuracy. In the external factors part, we use weather, holiday, day-of-week and time-of-day as external factors. Weather data including weather state (11 types, including sunny, cloudy, light snow, moderate snow, heavy snow, light rain, moderate rain etc.), temperature, wind speed and visibility. In our work, we crawl the weather data from a famous weather website[1]. Since the weather data at future time interval is unknown, we can use the predicted weather or exactly the weather of the last time. We divide 24 hours in each day into 2 parts as time-of-day: inactive hours (0:00-6:00) and active hours (6:00-24:00). The day-of-week is categorized to 7 types like Monday, Tuesday, Wednesday, Thursday etc.

In our architectue, the weather data and time metadata are fed into two-layer fully-connected neural network respectively, each of which has an embedding layer followed by the ReLU activation function and a layer to map the first layer result into the same shape as D_t. We define the output of the external component as O_{eh} and we use sigmoid fuction to fuse the output of the external component with the other output parts, which is defined as

$$\hat{D}_t = \sigma(O_{dc} + O_{eh}), \tag{12}$$

where $\hat{D}_t$ is the estimated passenger demand during the tth time interval.

During the training process of our FlowFlexDP, the goal is to minimize the mean squared error between the real and estimated passenger demand. The objective function of our architecture with α stands for all the learnable parameters is defined as

$$\min_{\alpha} \left\| D_t - \hat{D}_t \right\|_2^2. \tag{13}$$

The training process of our architecture FlowFlexDP is showed in Algorithm 1.

Algorithm 1. FlowFlexDP Training Algorithm

```
Input: Historical demand: ${D_1, D_2, \ldots, D_n}$
       Historical crowd outflow: ${C_1, C_2, \ldots, C_n}$
       Weather data: ${E_1, E_2, \ldots, E_n}$
       Time metadata: ${H_1, H_2, \ldots, H_n}$
       Length of half-hourly and daily sequences: $h$, $d$
Output: FlowFlexDP model with learned parameters
Initialize the training instances set $U \leftarrow 0$
for all available time intervals $t(2 \le t \le n)$ do
    $D_h = [D_{t-h}, D_{t-(h-1)}, \ldots, D_{t-1}]$
    $D_d = [D_{t-48d}, D_{t-48(d-1)}, \ldots, D_{t-48}]$
    $C_h = [C_{t-h}, C_{t-(h-1)}, \ldots, C_{t-1}]$
    $C_d = [C_{t-48d}, C_{t-48(d-1)}, \ldots, C_{t-48}]$
    Put $(\{D_h, D_d, C_h, C_d, E_{t-1}, H_t\}, S_t)$ into $U$
Initialize all the learnable parameters
repeat
    Randomly extract a batch of instances $U_b$ from $U$
    Minimize the loss function within $U_b$
until Convergence criterion met;
return FlowFlexDP model
```

6 Experimental Results

In this section, we present the extensive experimental results and evaluate our system for the prediction of passenger demand.

[1] http://www.weather.com.cn/

TABLE 3
The Statistics of Urban Region Clustering

item	value
Region Count	75
Max Area of Regions (km^2)	52.0
Min Area of Regions (km^2)	0.8
Average Area of Regions (km^2)	6.0
Max Count of Covered Cell Towers by Regions	255
Min Count of Covered Cell Towers by Regions	4
Average Count of Covered Cell Towers by Regions	53.0

6.1 Experiment Setting

6.1.1 Experimental Setup

We first cluster regions into 75 regions as Section 4 and extract passenger demand and crowd outflow sequence from our dataset. The statistics of regions are summarizes in Table 3. For spatial-temporal sequence like passenger demand and crowd outflow sequence, we use the Min-Max normalization to scale the data into the range [0,1]. For the external factors, we use one-hot coding to transform day-of-week, time-of-day, holidays, weather state into binary vectors, and use Min-Max normalization to scale the temperature, wind speed and visibility into the range [0,1]. We picked the first 65 percent of timestamps for the training data, the second 10 percent is chosen as the validation set and the rest 25 percent for testing. The prediction results are re-scaled to the normal values to calculate the prediction accuracy.

6.1.2 Evaluation Metric

We evaluate our architecture FlowFlexDP via Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}\left(D_t(r_i) - \hat{D}_t(r_i)\right)^2}, \tag{14}$$

where D and $\hat{D}$ are the ground truth and predicted passenger demand respectively. n is the total number of test predicted values, which is the number of regions in our experiment.

6.1.3 Parameters Setting

TensorFlow [29] is used to build our FlowFlexDP model. The batch size is set to 24 and the epoch is set to a fixed number. The length of half-hourly sequence $h \in \{3, 4, 5, 6, 7, 8, 9, 10\}$ and the length of daily sequence $d \in \{1, 2, 3, 4, 5, 6, 7, 8\}$, the layers number of Res-GCNN $L \in \{1, 2, 3, 4, 5\}$ are fine-tuned to achieve the best performance. Finally, we choose $h = 5$, $d = 4$, $L = 2$ as the final model, that means the four sequence components (passenger demand half-hourly sequence, passenger demand daily sequence, crowd flow half-hourly sequence and crowd outflow daily sequence) are respectively fed into deep neural networks which stack one layer GCNN, two layer ResGCNN, another one layer GCNN to encode the spatiotemporal dependencies. After that, the four encoding output are fused with global external factors to get the final prediction result. All experiments are run on a Centos machine with Intel Xeon E5-2620@2.10 GHz CPU and K40C 12 GB GPU. And we measure the inference time of DeepFlowFlex on the test data. The average inference time is 0.127s which is fast enough for real-time prediction.

6.2 Performance Evaluation

6.2.1 Baseline Models

We consider the following six baselines to compare with our architecture FlowFlexDP model:

- *HA*: The Historical Average model predicts the future passenger demand based on the average value of historical passenger demand in the corresponding periods. For example, the demand value of region i during 6:00 am-6:30 am is computed by the mean value of all historical demand from 6:00 am to 6:30 am in region i.
- *ARIMA*: The Autoregressive Integrated Moving Average model has been widely adopted in time series prediction. In an ARIMA(p, d, q) model, the future value of a variable is assumed to be a linear function of several past observations and random errors.
- *SARIMA*: the Seasonal Autoregressive Integrated Moving Average model can capture the seasonality in time series.
- *VAR*: Vector Auto-Regressive is a stochastic process model used to capture the linear interdependence among four types of sequences, which is a more advanced spatial-temporal model.
- *LSTM*: Long-Short Term Memory is a Recurrent Neural Network architecture that remembers values over arbitrary intervals. The LSTM model uses all the data, including historical passenger demand, crowd outflow, weather data, and time metadata to predict future passenger demand.
- *ANN*: The Artificial Neural Network employs all the data with look-back time window $K = 6$, including historical passenger demand intensity, crowd outflow, weather data, and time metadata to predict future passenger demand.

6.2.2 Performance Evaluation

We compare the performance of the 5 variants of our FlowFlexDP model and the 6 baseline models. For the 5 variants, GCNN is a model without using residual units and only using the passenger demand half-hourly sequence. ResGCNN is the model combines GCNN and residual learning. ResGCNN-D is the model further adds the passenger demand daily sequence. ResGCNN-DO further fuses the crowd outflow half-hourly and daily sequences. And our final model FlowFlexDP is a combination of ResGCNN with passenger demand half-hourly and daily sequences, crowd outflow hourly and daily sequences and external factors. Table 4 summarizes the performances of all models.

From the table, we can see that all of the 5 variants achieve better performance than the baseline models by at least 13.38 percent. Results of GCNN and ResGCNN show that applying residual units can make the prediction more accurate. The results of ResGCNN-D and ResGCNN-DO indicate the effectiveness of adding passenger demand daily sequence, crowd outflow hourly sequence and crowd outflow daily sequence. It is noteworthy that by incorporating crowd outflow, the prediction accuracy has

TABLE 4
Comparisions Between 6 Baseline Methods and 5 Variants

Model	RMSE
HA	17.28
ARIMA	11.52
SARIMA	10.33
VAR	9.16
ANN	8.95
LSTM	7.85
GCNN	6.80
ResGCNN	6.52
ResGCNN-D	6.11
ResGCNN-DO	5.93
FlowFlexDP (our model)	5.76

improved a lot. Then by accounting for the influence of external factors, our final FlowFlexDP model obtains an additional improvement of 2.87percent over ResGCNN-DO model, which does not fuse with the external factors influence. In our final FlowFlexDP model, we set half-hourly sequence length h to 5 and daily sequence length d to 4. Overall, our model FlowFlexDP obtains a better performance than the state-of-the-art demand prediction methods.

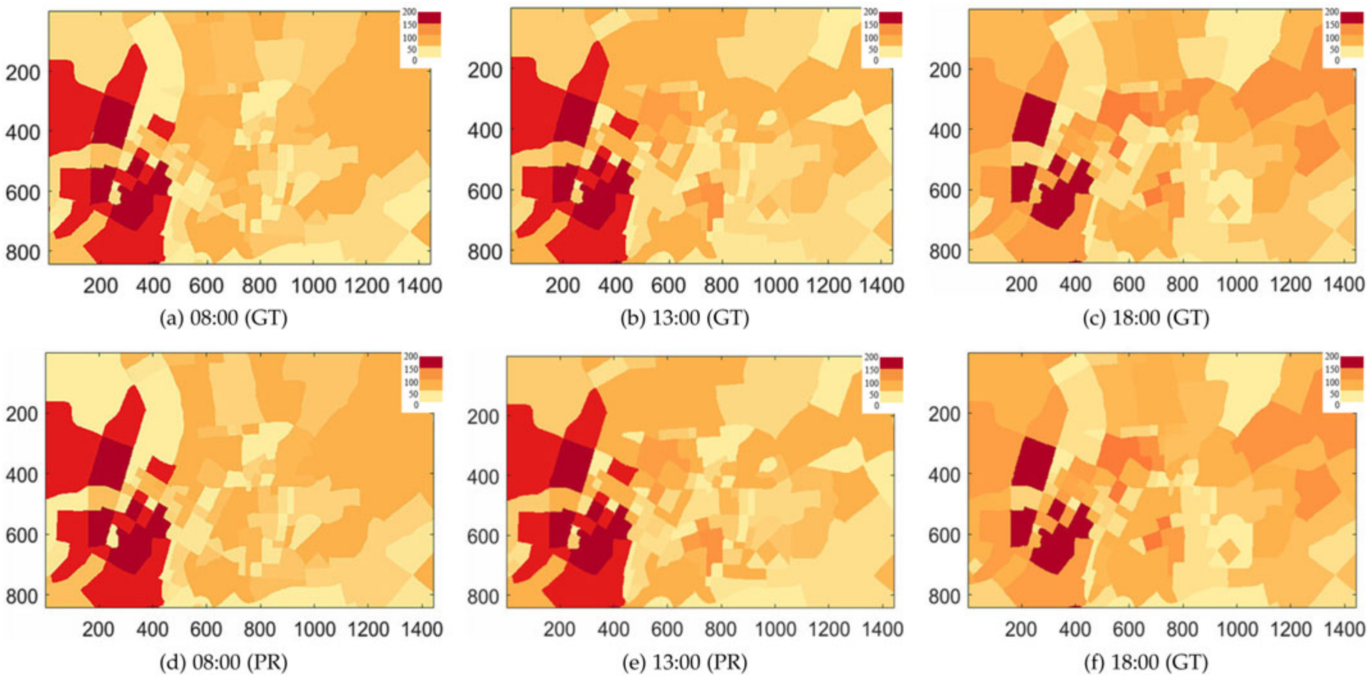
6.3 Visualization of Results

The visualization of the results is shown in Fig. 10, which depicts the heat maps of ground truth and the predicted result at two different times. We can see that our prediction results are very close to the real state, which means that our architecture FlowFlexDP can capture the spatial-temporal characteristics of the passenger demand and make more accurate prediction.

7 Related Work

7.1 Passenger Demand Prediction

Recently, taxi-passenger demand prediction has become one of the most important research topics in urban computing [30]. Passenger demand prediction relies on a predictive model and geographically labelled demand records through various localization technologies, such as GPS, WiFi positioning [31], [32], [33], cell tower positioning [34], etc. [2] proposed a model combined the Time-Varying Poisson Model, Weighted Time-Varying Poisson Model and auto-regressive integrated moving average (ARIMA) model to predict the number of services that will emerge at a given taxi stand, which aims to solve the best taxi stand choice problem after a passenger drop-off. [35] proposed an improved ARIMA model to predict the spatial-temporal passenger variation in hotspots and [3] implemented and compared three models, i.e., the Markov algorithm, Lempel-Ziv-Welch algorithm, and neural network. However, aforementioned studies work on GPS history data, which has a bias between the pick-up records and passenger demand. With the car-hailing service performing as pivotal public transportation, passenger demand prediction on car-hailing system also attracts researchers' attentions [36], [37], [38], [39], [40]. In order to predict passenger demand, [41], [42] presented non-parametric models, but they estimate passenger demand without any additional information such as weather and holidays. Besides temporal information of the passenger demand, [43] also uses the spatial and external information. [43] implemented the exponentially weighted moving-average (EWMA) model to recommend the top k hotspots for taxi drivers. However, their methods can only predict the hotpots rather than predicting an accurate demand number. In our work, we incorporate not only weather data, but also crowd outflow. [44] inferred the fine-grained and real time air quality information throughout a city based on the historical air quality and a variety of data sources including traffic



(a) 08:00 (GT)
(b) 13:00 (GT)
(c) 18:00 (GT)
(d) 08:00 (PR)
(e) 13:00 (PR)
(f) 18:00 (GT)

Fig. 10. Comparision of the ground truth(GT) and predicted passenger demand(PR) by FlowFlexDP.

flow, meteorology, human mobility, the structure of road networks and point of interests (POIs). Although this work used the relationships between the traffic flow and other data, it aims to infer the air quality instead of predicting the traffic flow. Some researchers aimed to predict passenger demand with deep learning models [5], [34]. Although similar with our work, these works are only applicable on fixed grid region partition or fail to capture the complex spatial dependency of passenger demand. [34] introduced residual learning into passenger demand prediction. However, the model uses fully connected layers to model the passenger demand, which is improper for learning the spatial-temporal dependencies of demand among regions in large-scale cities. All these methods are target to the number of pick-ups, but because of some passenger requests would be canceled or cannot match drivers successfully, the number of pick-ups cannot reflects the actual passenger demand in the real situation. [7] proposed a linear regression model and used features over space, time, meteorology and event domains to predict UOTD (Unit Original Taxi Demand), which reflects the complete original passenger demands for a given time and space. But to an extent, it is difficult to build a high-precision model if only uses the linear regression model. In our work, we propose a deep neural network to predict the complete original passenger demand on any shaped region, making it more practical.

7.2 Cellular Traffic Analysis

The cellular footprints of human activities have fostered research on the intersection between human and network dynamics [45] and enabled a set of applications, including human mobility modelling [46], land usage [9], data traffic engineering [47], public traffic [48], [49] and social interactions [50]. In this paper, we focus on investigating the passenger demand patterns via crowd flow extracted from cellular data, which advances the application of cellular footprints on passenger demand prediction, and paves a way towards a comprehensive understanding of the connection among mobile data traffic and human behaviors.

7.3 Deep Learning

CNN developed by [51] was used to learn the spatial correlation in network-wide traffic forecasting [52]. But it can only apply on regular grids, so graph convolutional neural network is designed for generalizing CNNs from regular grids to irregular domains [13]. Residual learning allows networks to have deeper structure [53]. Deep learning techniques have also been widely used in the traffic prediction area. [28] proposed a deep spatial-temporal residual network (ST-ResNet) to collectively forecast the inflow and outflow of crowds in every region of a city. However, the objective of this paper is to predict the taxi trajectories, which differs from the real passenger demand. To capture both spatial and temporal dependencies, [54] proposed the conv-LSTM network, which is applied to predict passenger demand [5]. But such model cannot capture very long-range temporal dependencies and is only applicable on regular grids. In this paper, we mainly propose employing the graph convolutional neural network and residual learning to model spatial-temporal passenger demand and crowd outflow data.

8 Conclusion

In this paper, We propose a deep learning model FlowFlexDP, which can solve the imbalance problem between supply and demand. Our model uses graph convolutional neural network and residual learning for predicting passenger demands on flexible region partition, based on weather, holidays, historical order and crowd flow extracted from cellular data. Evaluation results on a large-scale real data set show that our model outperforms existing models. As part of our future work, we consider to explore more data source such as the information in microblog or tweet and make full use of the cellular data to extract more useful data for making much more accurate prediction. Additionally, we expect to utilize more machine learning techniques to further capture the relationship of both the spatial-temporal data and external factors under the car-hailing service platform. Furthermore, it is also worth to design a user-friendly mobile interface for providing a convenient service for both passengers and drivers.

Acknowledgments

This work was supported in part by the National Key Research Plan under grant No. 2016YFC0700100, NSFC under Grants 61522110, 61332004, 61361166009, 61572366, 61602381, and 61472057. Jing Chu and Xu Wang are co-first authors of this work.

References

[1] X. M. Chen, M. Zahiri, and S. Zhang, “Understanding ridesplitting behavior of on-demand ride services: An ensemble learning approach,” *Elsevier Transp. Res. Part C: Emerg. Technol.*, vol. 76, pp. 51–70, 2017.

[2] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, “Predicting taxi–passenger demand using streaming data,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1393–1402, Sep. 2013.

[3] K. Zhao, D. Khryashchev, J. Freire, C. Silva, and H. Vo, “Predicting taxi demand at high spatial resolution: Approaching the limit of predictability,” in *Proc. IEEE Int. Conf. Big Data*, 2016, pp. 833–842.

[4] R. Wang *et al.*, “TaxiRec: Recommending road clusters to taxi drivers using ranking-based extreme learning machines,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 3, pp. 585–598, Mar. 2017.

[5] J. Ke *et al.*, “Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach,” *Transp. Res. Part C: Emerging Technol.*, vol. 85, pp. 591–608, 2017.

[6] H. Wei, Y. Wang, T. Wo, Y. Liu, and J. Xu, “ZEST: A hybrid model on predicting passenger demand for chauffeured car service,” in *Proc. ACM ICIKM*, 2016, pp. 2203–2208.

[7] D. Wang, W. Cao, J. Li, and J. Ye, “DeepSD: Supply-demand prediction for online car-hailing services using deep neural networks,” in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 243–254.

[8] H. Wang, F. Xu, Y. Li, P. Zhang, and D. Jin, “Understanding mobile traffic patterns of large scale cellular towers in urban environment,” in *Proc. Internet Measurement Conf.*, 2015, pp. 225–238.

[9] J. L. Toole, M. Ulm, M. C. González, and D. Bauer, “Inferring land use from mobile phone activity,” in *Proc. ACM SIGKDD Int. Workshop Urban Comput.*, 2012, pp. 1–8.

[10] C. Sarraute, P. Blanc, and J. Burroni, “A study of age and gender seen through mobile phone usage patterns in Mexico,” in *Proc. IEEE ASONAM*, 2014, pp. 836–843.

[11] H. Yang, C. W. Leung, S. C. Wong, and M. G. Bell, “Equilibria of bilateral taxi–customer searching and meeting on networks,” *Transp. Res. Part B: Methodol.*, vol. 44, no. 8, pp. 1067–1083, 2010.

[12] X. Xu, B. Su, X. Zhao, Z. Xu, and Q. Z. Sheng, “Effective traffic flow forecasting using taxi and weather data,” in *Proc. Int. Conf. Adv. Data Mining Appl.*, 2016, pp. 507–519.

[13] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[15] F. Calabrese, L. Ferrari, and V. D. Blondel, "Urban sensing using mobile phone network data: A survey of research," *ACM Comput. Surv.*, vol. 47, no. 2, 2015, Art. no. 25.

[16] I. Leontiadis, A. Lima, H. Kwak, R. Stanojevic, D. Wetherall, and K. Papagiannaki, "From cells to streets: Estimating mobile paths with cellular-side data," in *Proc. 10th ACM Int. Conf. Emerg. Netw. Experiments Technologies*, 2014, pp. 121–132.

[17] XK72, "Charles is an HTTP proxy / HTTP monitor / Reverse Proxy," 2017. [Online]. Available: https://www.charlesproxy.com/

[18] Dark Sky, "Weather data retrieved from https://planet.osm.org," 2017. [Online]. Available: https://darksky.net/dev

[19] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, Art. no. 92.

[20] M. X. Hoang, Y. Zheng, and A. K. Singh, "FCCF: Forecasting citywide crowd flows based on big data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2016, pp. 1–10.

[21] S. Gao, K. Janowicz, and H. Couclelis, "Extracting urban functional regions from points of interest and human activities on location-based social networks," *Trans. GIS*, vol. 21, no. 3, pp. 446–467, 2017.

[22] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org," 2017. [Online]. Available: https://www.openstreetmap.org

[23] R. M. Haralock and L. G. Shapiro, *Computer and Robot Vision*. Reading, MA, USA: Addison-Wesley, 1991.

[24] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Neural Inf. Process. Syst.*, 2002, pp. 849–856.

[25] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *J. Roy. Statist. Soc., Series C (Appl. Statist.)*, vol. 28, no. 1, pp. 100–108, 1979.

[26] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[27] M. Meila and J. Shi, "Learning segmentation by random walks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 873–879.

[28] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[29] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

[30] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 89–98.

[31] C. Wu, Z. Yang, and C. Xiao, "Automatic radio map adaptation for indoor localization using smartphones," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 517–528, Mar. 2018.

[32] Z. Yin, C. Wu, Z. Yang, and Y. Liu, "Peer-to-peer indoor navigation using smartphones," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1141–1153, May 2017.

[33] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 444–457, Feb. 2015.

[34] H. Wu, W. Sun, B. Zheng, L. Yang, and W. Zhou, "CLSTERS: A general system for reducing errors of trajectories under challenging localization situations," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 3, 2017, Art. no. 115.

[35] X. Li *et al.*, "Prediction of urban human mobility using large-scale taxi traces and its applications," *Front. Comput. Sci.*, vol. 6, no. 1, pp. 111–121, 2012.

[36] D. Zhang, T. He, S. Lin, S. Munir, and J. A. Stankovic, "Dmodel: Online taxicab demand model from big sensor data in a roving sensor network," in *Proc. IEEE Int. Congr. Big Data*, 2014, pp. 152–159.

[37] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2588–2595.

[38] J. Xu, R. Rahmatizadeh, L. Bölöni, and D. Turgut, "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2572–2581, Aug. 2017.

[39] F. Rodrigues, I. Markou, and F. C. Pereira, "Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach," *Inf. Fusion*, vol. 49, pp. 120–129, 2019.

[40] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, "Contextualized spatial–temporal network for taxi origin-destination demand prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3875–3887, Oct. 2019.

[41] A. Afian, A. Odoni, and D. Rus, "Inferring unmet demand from taxi probe data," in *Proc. IEEE 18th Int. Conf.*, 2015, pp. 861–868.

[42] Y. Li, L. Jian, Z. Lin, and Z. Yi, "Taxi booking mobile app order demand prediction based on short-term traffic forecasting," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2634, pp. 57–68, 2017.

[43] Z. Kai, Z. Feng, S. Chen, K. Huang, and G. Wang, "A framework for passengers demand prediction and recommendation," in *Proc. IEEE Int. Conf. Serv. Comput.*, 2016, pp. 340–347.

[44] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 1436–1444.

[45] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-scale mobile traffic analysis: A survey," *IEEE Commun' Surveys Tut.*, 2016, 124–161.

[46] C. Song, T. Koren, P. Wang, and A.-L. Barabási, "Modeling the scaling properties of human mobility," *Nature Physics*, vol. 6, no. 10, 2010, pp. 818–823.

[47] X. Wang, Z. Zhou, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," in *Proc. IEEE 25th Int. Conf. Netw. Protocols*, 2017, pp. 1–10.

[48] P. Zhou, Y. Zheng, and M. Li, "How long to wait? predicting bus arrival time with mobile phone based participatory sensing," *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1228–1241, Jun. 2014.

[49] D. Zhang, J. Huang, Y. Li, F. Zhang, C. Xu, and T. He, "Exploring human mobility with multi-source data at extremely large metropolitan scales," in *Proc. 20th ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 201–212.

[50] X. Zhang *et al.*, "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 54–67, Firstquarter 2016.

[51] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," *Shape, Contour and Grouping in Computer Vision*, Springer, Berlin, Germany, pp. 823–823, 1999.

[52] Q. Chen, X. Song, H. Yamada, and R. Shibasaki, "Learning deep representation from big and heterogeneous data for traffic accident inference," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 338–344.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[54] X. Shi, 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 *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

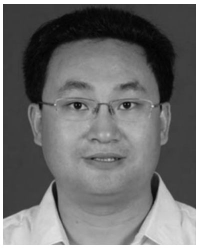
Jing Chu (Student Member, IEEE) received the BE degree from the School of Software, Harbin Institute of Technology, in 2016, she is currently working toward the master's degree with the School of Software, Tsinghua University. She is a member of the Tsinghua National Lab for Information Science and Technology. Her research interest is in mobile computing.

Xu Wang (Student Member, IEEE) received the BE degree from the School of Software, Tsinghua University, in 2015, he is currently working toward the PhD degree with the School of Software, Tsinghua University. He is a member of the Tsinghua National Lab for Information Science and Technology. His research interests include wireless networks and mobile computing.

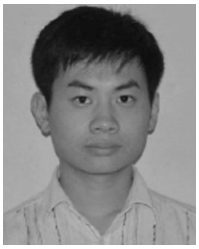
Kun Qian (Student Member, IEEE) received the BE degree from the School of Software, Tsinghua University, in 2014, he is currently working toward the PhD degree with the School of Software, Tsinghua University. He is a member of the Tsinghua National Lab for Information Science and Technology. His research interests include wireless networks and mobile computing.

Lina Yao (Member, IEEE) received the PhD degree in computer science from the University of Adelaide, Australia. She is currently a lecturer with the School of Computer Science and Engineering, UNSW. Her research interests include the Internet of Things, information filtering and recommending, and human activity recognition. She is the author of more than 60 publications. She is a member of the ACM.

Fu Xiao (Member, IEEE) received the PhD degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2007. He is currently a professor and also a PhD supervisor with the School of Computer, Nanjing University of Posts and Telecommunications, Nanjing. His main research interest is wireless sensor networks. He is a member of the IEEE Computer Society and the Association for Computing Machinery.



Jianbo Li received the BS and MS degrees in computer science from Qingdao University, China, in 2002 and 2005, respectively, and the PhD degree in computer science from the University of Science and Technology of China, in 2009. He is currently a professor with the Computer Science and Technology College, Qingdao University. His research interests include opportunistic networks, data offloading techniques, and intelligent city.



Zheng Yang (Senior Member, IEEE) received the BE degree in computer science from Tsinghua University, in 2006, and the PhD degree in computer science from the Hong Kong University of Science and Technology, in 2010. He is currently an associate professor with Tsinghua University. His main research interests include wireless ad-hoc/sensor networks, and mobile computing. He is a member of the ACM.