Passenger Demand Prediction with Cellular Footprints

Jing Chu*, Kun Qian*, Xu Wang*, Lina Yao†, Fu Xiao‡, Jianbo Li§, Xin Miao* and Zheng Yang*
*School of Software, Tsinghua University
†School of Computer Science and Engineering, The University of New South Wales
‡School of Computer Science, Nanjing University of Posts and Telecommunications
§College of Computer Science and Techonology, Qingdao University
j-zhu16@mails.tsinghua.edu.cn, qiank10@gmail.com, xu-wang15@mails.thu.edu.cn,
theresa0125@gmail.com, xiaof@njupt.edu.cn, lijianbo@188.com, machine.miao@gmail.com, hmilyyz@gmail.com

***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 paper, 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 DiDi Chuxing from cellular data shows that FlowFlexDP accurately predicts passenger demand and outperforms the state-of-the-art demand prediction methods.**

I. Introduction

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 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, 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

978-1-5386-4281-8/18/$31.00 ©2018 IEEE

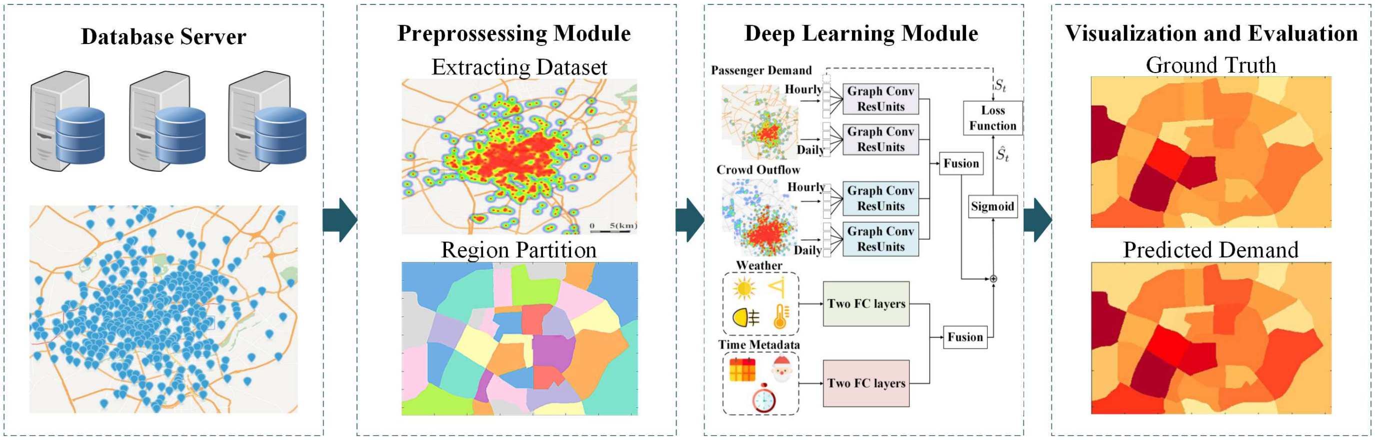


Fig. 1: System Architecture. The system consists of 4 modules: database module, pre-processing module, deep learning module, and visualization and evaluation module.

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 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 GCNN [12] to adapt to flexible region partition, and uses residual network [13] 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.

TABLE I: Corresponding results of ordering taxis

Action	URL(api.diditaxi.com.cn/)
Start a 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

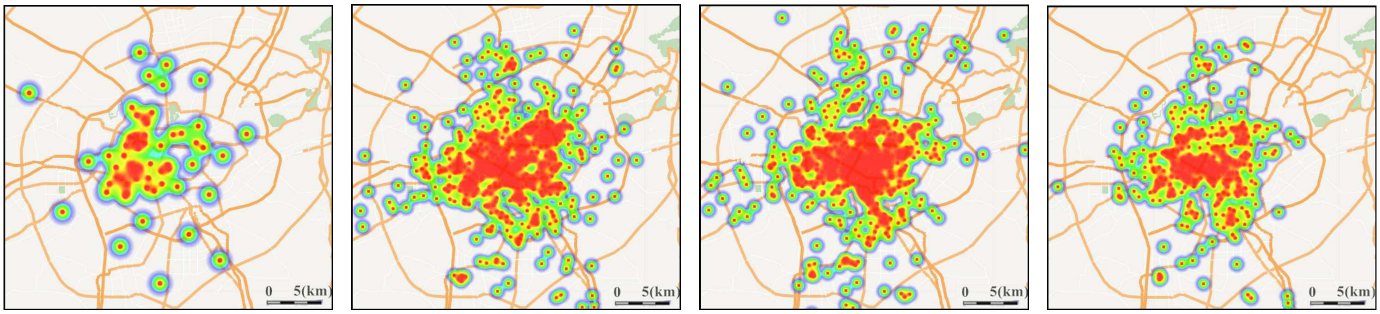
- We extract a large scale real data set of DiDi Chuxing from cellular traffic data, and conduct extensive experiments. The results show that FlowFlexDP achieves a more accurate prediction accuracy. By considering crowd flow data, and accommodating flexible region partition, FlowFlexDP outperforms existing previous models.

II. System Overview and Data Source

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

A. 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 retrival and aggregation services for fast pre-processing. In the pre-processing module we clean the data and extract the passenger 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 evalute the performance of the system and generate real-time heatmaps of passenger demand to advice drivers where to pick up passengers.



(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.

B. Data Source

1) Cellular Data: Our dataset was collected by a major cellular carrier in a big city of China. 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 connnected cell tower ID, App ID, URL, 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. 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.

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 [14]. However, a location accuracy of 70m is achievable with the coverage information of cell towers [15].

In order to understand the meaning behind the URLs in our data set, we did a field experiment to capture the HTTP and SSL/HTTPS header by some HTTP proxy tool (*i.e.*, Charles [16]) when taking DiDi taxis and other types of cars (e.g. DiDi Express, DiDi Premier). The collected URLs are matched with the volunteer's user behaviour. The corresponding results of ordering a taxi are shown in Table I. 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.

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 [17]. The overview of the weather data is illustrated in Table II.

TABLE II: 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]

III. Problem Formulation

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

Definition 1 (Passenger Demand): Let $R = r_1, r_2, \cdots, r_N$ 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 at region r_i. And $D_t = \{D_t(r_i)|i = 1, \dots, N\}$ represents the sequence of passenger demand of all regions at time t.

Definition 2 (Crowd Outflow): Crowd Outflow [18] of a region means the number of people leaving the region. Similarly, we use $P_t(r_i)$ to denote the set of people located inside the region r_i at a given time interval $[t, t + \Delta t]$. 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 $C_t = \{C_t(r_i)|i = 1, \dots, N\}$ denotes the sequence of outflow of all regions at time t.

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

IV. 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. To extract semantic regions, we design

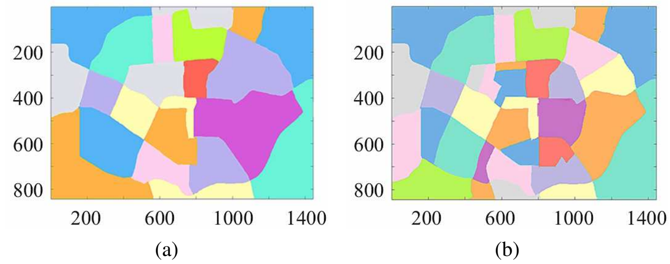


Fig. 3: Process of region partition: (a) entire city partitioned by the primary roads; (b) the finer-grained partition.

our region partition algorithm partitioning the whole city into regions by road network.

Initially, we partition the entire city into blocks by the primary roads. Particularly, we extract the primary road network from OpenStreetMap [19] and convert the map to a binary image where roads are represented by black lines. Further, we apply "bwlabel" algorithm [20] to detect the connected components as partition result. Fig. 3a shows the primary region partition and we get 22 regions.

Partition with only primary road is insufficient because the center regions take a great percentage of passenger demands. Besides, finer-grained partition is more helpful to drivers. Accordingly, we further partition the city by the main secondary roads if the passenger demand of any block is too high. In our work, we assess the passenger demand of each block in four different time, i.e. weekday morning rush time T_{dm}, weekday evening rush time T_{de}, weekend morning rush time T_{em}, and weekend evening rush time T_{ee}. We rank the passenger demand of the four time and choose the first 3 blocks respectively. If the latitude span or longtitude span of the selected block is larger than a certain distance d_{span}, we partition it by the secondary road in it. The finer-grained partition is shown as Fig. 3b and the number of regions is 35.

V. 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. 4 illustrates the architecture of FlowFlexDP.

In our model, historical passenger demand, historical crowd outflow and global external factors are fed as the input. Historical passenger demand is decomposed into passenger demand hourly sequence and passenger demand daily sequence. Similarly, historical crowd outflow is decomposed into crowd flow hourly sequence and crowd outflow daily sequence. Each of the sequence components is fed into deep neural networks, which are made up of GCNN 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.

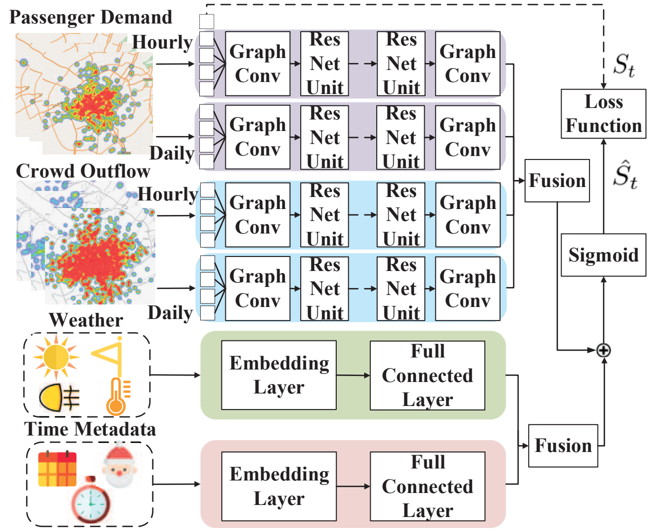


Fig. 4: The architecture of FlowFlexDP

A. Graph Convolutional Neural Network

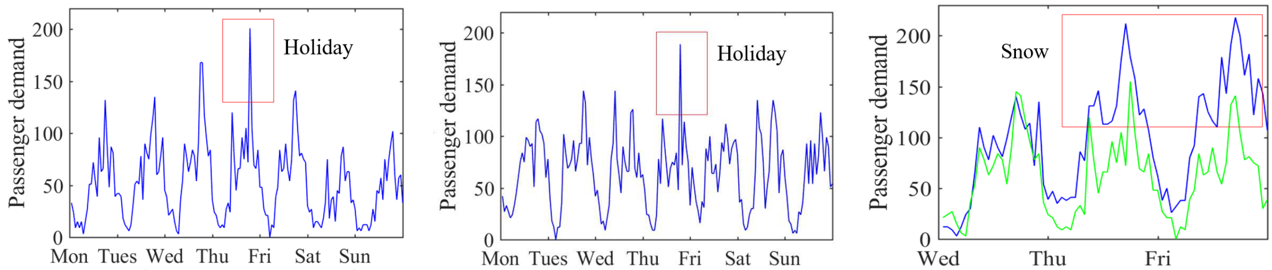
As we know, the passenger demand in one region is not only determined by the context of the region, but also influenced by the context in other regions [11]. And the dependency between nearby regions is stronger than distant regions. To model the spatial dependency, we apply GCNN in our network architecture. GCNN is designed for generalizing CNNs from low-dimensional regular grids where image, video and speech are represented, to high-dimensional irregular domains to learn the local, stationary and compositional spatio-temporal features. Formally, we model the city regions as a weighted graph $G = (V, E, W)$, where $V = \{r_i | i = 1, 2, \cdots, N\}$ is a finite set of regions. $E = \{(r_i, r_j) | r_i$ and r_j are adjacent$\}$ is a set of edges. W is the adjacency matrix. The normalized graph Laplacian L of G is defined as:

$$L = I_n - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \quad (1)$$

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:

$$y = g_\theta(L)x = g_\theta(U \Lambda U^T)x = U g_\theta(\Lambda) U^T x \quad (2)$$

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



(a) Office Region

(b) Residential Region

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

Fig. 5: 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.

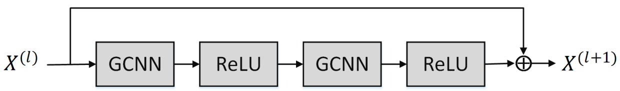


Fig. 6: The residual unit of GCNN.

Furthermore, computing the eigendecomposition of L is prohibitively expensive for large graphs. To overcome this problem, a polynomial filter is used and defined as:

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k \tag{3}$$

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.

B. 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 [13] 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:

$$x_{l+1} = x_l + F(x_l) \tag{4}$$

where x_{l+1} and x_l are the input and output of the l-th unit in the network, F is the mapping function. Fig. 6 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.

C. Hourly and Daily Sequences Fusion

ResGCNN can capture the spatial dependencies, but it fails to capture the temporal denpendencies. In order to learn the spatio-temporal dependencies, we divide the passenger demand sequences into two parts, the hourly sequence and daily sequence. For hourly sequence, there is no doubt that the passenger demand of a time is affected by the previous hours. As for daily sequence, Fig. 5a and Fig. 5b 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 one hour time interval. The corresponding hourly sequence at t is $[D_{t-h}, D_{t-(h-1)}, \cdots, D_{t-1}]$ and daily sequence at t is $[D_{t-T\cdot d}, D_{t-T\cdot(d-1)}, \cdots, D_{t-T}]$,where h is the length of hourly time intervals and d is the length of daily sequence, T=24 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 Fig. 5a and Fig. 5b, 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 hourly and daily sequences may have different influence

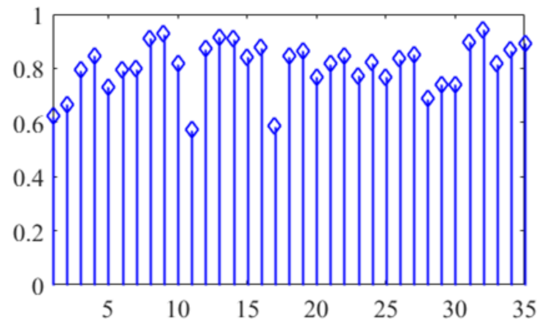


Fig. 7: The cross correlation coefficient of passenger demand sequence and crowd outflow sequence in each region.

degrees. We fuse the two components with parametric-matrix-based fusion [21], which is defined as

$$D_o = W_{ho} \circ D_{ho} + W_{do} \circ D_{do} \tag{5}$$

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 hourly and daily sequences.

D. Crowd Outflow Sequences Fusion

Fig. 7 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 hourly sequence and daily sequence of crowd outflow at time t as $[C_{t-h}, C_{t-(h-1)}, \cdots, C_{t-1}]$ and $[C_{t-T\cdot d}, C_{t-T\cdot(d-1)}, \cdots, C_{t-T}]$, respectively.

For these two kinds of sequences, we feed them into another two deep neural networks, respectively. The output of crowd outflow hourly sequence is C_{ho} and for daily sequence is C_{do}. These sequences are fused with parametric-matrix-based fusion [21]. This fusion method uses Hadamard product to element-wise multiply learnable parameter and output vector, which is defined as

$$C_o = W_{hc} \circ C_{ho} + W_{dc} \circ C_{do} \tag{6}$$

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

$$O_{dc} = C_o + D_o \tag{7}$$

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

E. External Factors Fusion

Fig. 5a shows the influence of daytype (*i.e.*, weekday or weekend), Fig. 5c illustrates that in comparison with sunny days, the demand on snowy days increase sharply, and Fig. 5b 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 etc.), temperature, wind speed and visibility. 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, 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

$$\hat{D}_t = \sigma(O_{dc} + O_{eh}) \tag{8}$$

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

$$\min_{\alpha} \left\| D_t - \hat{D}_t \right\|_2^2 \tag{9}$$

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

VI. Experimental Results

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

A. Experiment Setting

1) Experimental Setup: For spatio-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% of timestamps for the training data, the second 10% is chosen as the validation set and the rest 25% for testing. The prediction results are re-scaled to the normal values to calculate the prediction accuracy.

2) Parameters Setting: TensorFlow [22] 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 hourly sequence

Algorithm 1: FlowFlexDP Training Algorithm

Input: Historical demand: $\{D_1, D_2, \cdots, D_n\}$
Historical crowd outflow: $\{C_1, C_2, \cdots, C_n\}$
Weather data: $\{E_1, E_2, \cdots, E_n\}$
Time metadata: $\{H_1, H_2, \cdots, H_n\}$
Length for hourly and daily sequences: h, d
Output: FlowFlexDP model with learned parameters

Initialize the trianing instances set $U \leftarrow 0$
for *all available time intervals* $t(2 \leq t \leq n)$ **do**
    $D_h = [D_{t-h}, D_{t-(h-1)}, \cdots, D_{t-1}]$
    $D_d = [D_{t-24d}, D_{t-24(d-1)}, \cdots, D_{t-24}]$
    $C_h = [C_{t-h}, C_{t-(h-1)}, \cdots, C_{t-1}]$
    $C_d = [C_{t-24d}, C_{t-24(d-1)}, \cdots, C_{t-24}]$
    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 objective function within U_b
until *Convergence citerion met*
return FlowFlexDP model

$h \in \{3, 4, 5, 6, 7, 8\}$ and the length of daily sequence $d \in \{1, 2, 3, 4, 5, 6, 7, 8\}$.

B. Performance Evaluation

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

$$RMSE = \sqrt{\frac{1}{n} \sum_i \left(D_t(r_i) - \hat{D}_t(r_i)\right)^2} \tag{10}$$

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

2) 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 6:00am-7:00am is computed by the mean value of all historical demand from 6:00am to 7:00am 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 which 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 spatio-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.

TABLE III: Comparisions between 6 baseline methods and 5 variants.

Model	RMSE
HA	23.21
ARIMA	16.44
SARIMA	15.29
VAR	13.72
ANN	12.18
LSTM	10.35
GCNN	9.16
ResGCNN	8.97
ResGCNN-D	8.71
ResGCNN-DO	7.13
FlowFlexDP (our model)	7.02

3) 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 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 hourly and daily sequences. And our final model FlowFlexDP is a combination of ResGCNN with passenger demand hourly and daily sequences, crowd outflow hourly and daily sequences and external factors. Table III 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 12.99%. 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 improved a lot. Then by accounting for the influence of external factors, our final FlowFlexDP model obtain an additional improvement of 1.6% over ResGCNN-DO model which does not fuse with the external factors influence. Overall, our model FlowFlexDP

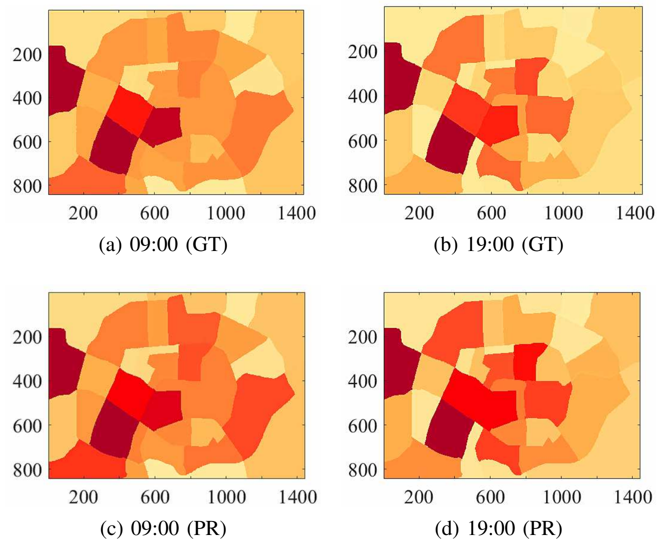


Fig. 8: Comparision of the ground truth(GT) and predicted passenger demand(PR) by FlowFlexDP.

obtains a better performance than the state-of-the-art demand prediction methods.

C. Visualization of Results

The visualization of the results is shown in Fig. 8, 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 spatio-temporal characteristics of the passenger demand and make more accurate prediction.

VII. Related Work

A. Passenger demand prediction

Recently, taxi-passenger demand prediction has become one of the most important research topics in urban computing [23]. Passenger demand prediction relies on a predictive model and geographically labelled demand records through various localization technologies, such as GPS, WiFi positioning [24]–[26], cell tower positioning [27], etc. [2] proposed a model combined the Poisson Model and ARIMA 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 [28]. In order to predict passenger demand, [29], [30] presented non-parametric models, but they estimate passenger demand without any additional information such as weather and holidays. In our work, we incorporate not only weather data, but also crowd outflow. Some researchers aimed to predict passenger demand with deep learning models [5], [27]. 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. [27] 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 spatio-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.

B. Cellular traffic analysis

The cellular footprints of human activities have fostered research on the intersection between human and network dynamics [31] and enabled a set of applications, including human mobility modelling [32], land usage [9], data traffic engineering [33], public traffic [34], [35] and social interactions [36]. 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.

C. Deep learning

CNN developed by [37] was used to learn the spatial correlation in network-wide traffic forecasting [38]. But it can only apply on regular grids, so graph convolutional neural network is designed for generalizing CNNs from regular grids to irregular domains [12]. Residual learning allows networks to have deeper structure [39]. And to capture both spatial and temporal dependencies, [40] 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 spatio-temporal passenger demand and crowd outflow data.

VIII. Conclusion

We propose a deep learning model FlowFlexDP, which uses GCNN 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.

IX. Acknowledgement

This work is supported in part by the National Key Research Plan under grant No. 2016YFC0700100, NSFC under grant 61522110, 61332004, 61361166009, 61572366, 61602381, 61472057.

References

[1] X. M. Chen, M. Zahiri, and S. Zhang, "Understanding ridesplitting behavior of on-demand ride services: An ensemble learning approach," *Elsevier Transportation Research Part C: Emerging Technologies*, 2017.

[2] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 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 *Proceedings of IEEE International Conference on Big Data*, 2016.

[4] R. Wang, C.-Y. Chow, Y. Lyu, V. Lee, S. Kwong, Y. Li, and J. Zeng, "Taxirec: recommending road clusters to taxi drivers using ranking-based extreme learning machines," in *Proceedings of ACM SIGSPATIAL*, 2015.

[5] J. Ke, H. Zheng, H. Yang *et al.*, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *arXiv preprint arXiv:1706.06279*, 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 *Proceedings of ACM ICIKM*, 2016.

[7] D. Wang, W. Cao, J. Li, and J. Ye, "Deepsd: supply-demand prediction for online car-hailing services using deep neural networks," in *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 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 *IMC*, 2015.

[9] J. L. Toole, M. Ulm, M. C. González, and D. Bauer, "Inferring land use from mobile phone activity," in *Proceedings of the ACM SIGKDD international workshop on urban computing*, 2012.

[10] C. Sarraute, P. Blanc, and J. Burroni, "A study of age and gender seen through mobile phone usage patterns in mexico," in *Proceedings of IEEE ASONAM*, 2014.

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

[12] 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.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[14] F. Calabrese, L. Ferrari, and V. D. Blondel, "Urban sensing using mobile phone network data: a survey of research," *Acm computing surveys (csur)*, vol. 47, no. 2, p. 25, 2015.

[15] 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 *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM, 2014, pp. 121–132.

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

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

[18] 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.

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

[20] R. M. Haralock and L. G. Shapiro, *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., 1991.

[21] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction." in *AAAI*, 2017, pp. 1655–1661.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[23] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 89–98.

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

[25] Z. Yin, C. Wu, Z. Yang, and Y. Liu, "Peer-to-peer indoor navigation using smartphones," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1141–1153, 2017.

[26] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 444–457, 2015.

[27] H. Wu, W. Sun, B. Zheng, L. Yang, and W. Zhou, "Clsters: A general system for reducing errors of trajectories under challenging localization situations," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 115, 2017.

[28] 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 *Big Data (BigData Congress), 2014 IEEE International Congress on*. IEEE, 2014, pp. 152–159.

[29] A. Afian, A. Odoni, and D. Rus, "Inferring unmet demand from taxi probe data," in *ITSC, 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 861–868.

[30] Y. Li, J. Lu, L. Zhang, and Y. Zhao, "Taxi booking mobile app order demand prediction based on short-term traffic forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2634, pp. 57–68, 2017.

[31] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-scale mobile traffic analysis: a survey," *IEEE Communications Surveys & Tutorials*, 2016.

[32] C. Song, T. Koren, P. Wang, and A.-L. Barabási, "Modeling the scaling properties of human mobility," *arXiv preprint arXiv:1010.0436*, 2010.

[33] X. Wang, Z. Zhou, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," in *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, 2017, pp. 1–10.

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

[35] 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 *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 201–212.

[36] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[37] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," *Shape, contour and grouping in computer vision*, pp. 823–823, 1999.

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

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[40] 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.