A General Framework for Unmet Demand Prediction in On-demand Transport Services

Wengen Li, Jiannong Cao, *Fellow, IEEE,* Jihong Guan, Shuigeng Zhou, *Member, IEEE,* Guanqing Liang, Winnie K.Y. So, Michal Szczecinski

***Abstract*—Emerging on-demand transport services, such as Uber and GoGoVan, usually face the dilemma of demand-supply imbalance, meaning that the spatial distributions of orders and drivers are imbalanced. Due to such imbalance, much supply resource is wasted while a considerable amount of order demand cannot be met in time. To address this dilemma, knowing the unmet demand in near future is of high importance for service providers because they can dispatch their vehicles in advance to alleviate the impending demand-supply imbalance. Therefore, we develop a general framework for predicting the unmet demand in future time slots. Under this framework, we first evaluate the predictability of unmet demand in on-demand transport services and find that unmet demand is highly predictable. Then, we extract both static and dynamic urban features relevant to unmet demand from datasets in multiple domains. Finally, multiple prediction models are trained to predict unmet demand by using the extracted features. As demonstrated via experiments, the proposed framework can predict unmet demand in on-demand transport services effectively and flexibly.**

***Index Terms*—On-demand transport service, unmet demand, predictability, prediction model**

I. Introduction

As one successful representative of online to offline (O2O) business paradigms, on-demand transport services have received a rapid proliferation in recent years due to the remarkable flexibility. For example, on-demand taxi services, such as Uber[1] and DiDi[2], are widely used in many cities around the world, which greatly improves users' travelling experience. Another example is on-demand cargo transport service, e.g., GoGoVan[3], which grows fast and makes it amazingly convenient for users to transport their goods and parcels.

Fig. 1 illustrates the general operation flow in on-demand transport services. There are three parties involved, i.e., users, drivers and service center. First, users issue orders (demand) to the service center and service center dispatches these orders to available drivers (supply). Then, drivers decide to accept or reject the received orders autonomously according to their preferences. If one order is accepted, the driver information will be sent to the corresponding user. Finally, drivers finish orders in the physical world.

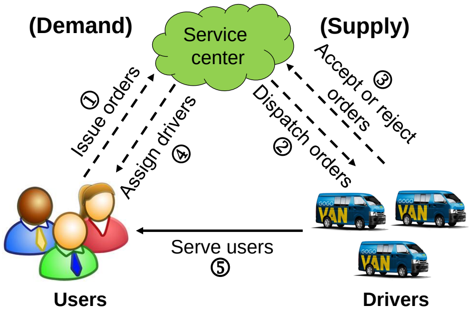


Fig. 1. General operation flow in on-demand transport services.

In practice, the spatial imbalance between order demand and available drivers is one dilemma for many on-demand transport service companies. In this case, order demand from users and service supply provided by drivers cannot geographically match with each other. Fig. 2 illustrates the total order demand and the unmet demand (i.e., the orders that are not accepted by drivers) of GoGoVan in a region of Hong Kong over hourly time slots. Obviously, considerable numbers of orders are not accepted by drivers. If such imbalance is serious, neither users' demand can be satisfied in time nor drivers can get enough orders to ensure a high income.

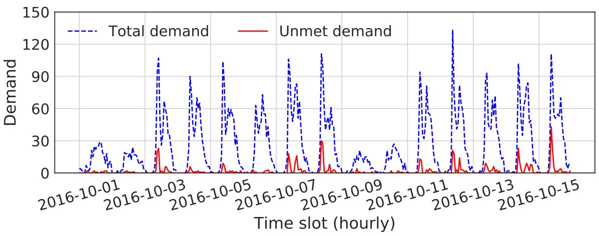


Fig. 2. The total demand and unmet demand for GoGoVan in a region of Hong Kong.

To solve the dilemma of demand-supply imbalance, knowing the unmet demand in near future is important because service providers can dispatch their vehicles in advance to alleviate the potential imbalance. Therefore, we aim to predict the unmet demand in future time slots for on-demand transport

This work was partially supported by the RGC General Research Fund under Grant PolyU 152199/17E, the funding for Project of Strategic Importance provided by the Hong Kong Polytechnic University (Project Code: 1-ZE26), the NSFC Key Grant with Project No. 61332004, the National Natural Science Foundation of China (U1636205, 61772367), and the Program of Science and Technology Innovation Action of Science and Technology Commission of Shanghai Municipality (STCSM) (No. 17511105204).

W. Li, J. Cao and G. Liang are with the Department of Computing, Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: {cswgli, csjcao, csgliang}@comp.polyu.edu.hk).

J. Guan is with the Department of Computer Science and Technology, Tongji University, Shanghai, China (e-mail: jhguan@tongji.edu.cn).

S. Zhou is with Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, Shanghai, 200433, China (e-mail: sgzhou@fudan.edu.cn).

W. So and M. Szczecinski are with Department of Data Analytics, GoGoTech Company, Hong Kong SAR, China (e-mail: bwsky8@gmail.com, michal@gogotech.hk).

[1] https://www.uber.com/

[2] http://www.didichuxing.com/en/

[3] https://www.gogovan.com.hk/en/

services. With the predicted results, available drivers can be dispatched to the regions with large unmet demand, thus improving users' satisfaction and helping drivers to get more orders.

Though demand prediction has been well studied in traditional taxi services, unmet demand prediction remains to be an open issue with limited research, which will be discussed in Section II in detail. Compared with demand prediction, unmet demand prediction is a more difficult problem because it depends on both demand from users and supply provided by drivers. To predict unmet demand in on-demand transport services effectively, we propose a general solution framework which extracts both static and dynamic urban features from data in multiple domains and trains prediction models using the extracted features to predict unmet demand in future time slots. To summarize, we make the following contributions in this work:

- We formally formulate the unmet demand prediction problem in on-demand transport services (Section III) and evaluate the predictability of unmet demand using the entropy of unmet demand record sequence in history (Section IV).
- We devise a general framework to extract both static and dynamic urban features from data in multiple domains to cover as many as possible the factors relevant to unmet demand prediction. Particularly, we solve the sparsity issue in extracting dynamic traffic features (Sections V & VI).
- We train multiple feature-based prediction models with the extracted features to predict the unmet demand in different regions and demonstrate the effectiveness of our proposed framework via extensive experiments. (Sections VII & VIII).

In addition, we review the related work and conclude the paper in Sections II and IX, respectively.

II. Related Work

In this section, we review the existing studies on *demand analysis* and *unmet demand analysis* in transport services. Most of these studies focus on traditional taxi services in which drivers pick up passengers within the visual range rather than accepting orders via mobile applications. In addition, our solution will utilize the discovery about feature selection in [1] to guide the model training.

A. Demand Analysis

The real demand in transport services refers to the number of passengers who need a car. However, most existing studies on demand analysis regard the number of pickups as the demand due to the difficulty of obtaining the number of passengers failing to find a car. These studies can be roughly classified into the following categories:

Short-term demand prediction. Approaches in this category predict the demand within a short future time span. These approaches can be further divided into time series-based approaches and model-based approaches [2]. Time series-based approaches usually use time series analysis techniques, e.g., ARIMA and Holt-Winters [3], to achieve demand prediction. Specifically, Moreira-Matias et al. [4], [5] predicted the demand at a given taxi stand within a short-time horizon in future by using ARIMA model and Sliding-Window Ensemble Framework. Li et al. [6] predicted the number of orders in future 60 minutes at taxi hotspots rather than taxi stands via ARIMA model. In model-based approaches, the demand is predicted using advanced models, such as linear regression model [7], neural network model [8], [9], and probabilistic model [10], [11]. Approaches based on these models can achieve satisfying performance provided that the models are carefully designed and trained.

Demand hotspot identification. These approaches aim to identify the hot regions that have large demand. To this end, clustering algorithms, such as DBSCAN [12], are used to cluster historical orders. Specifically, Liu et al. [13] devised a density-based clustering algorithm to discover demand hot spots based on a taxi dataset. To solve the efficiency issue of density-based clustering algorithm when applied on large datasets, Zhang et al. [14] introduced Grid and KD-tree [15] to speed-up the computation process. In addition, the contexts, such as weather and time, are considered in [16] to cluster orders. Another work [17] proposed a demand hotspot prediction framework to generate recommendations for taxi drivers by using spatio-temporal clustering over orders.

Demand pattern discovering. In addition to short-term demand prediction and demand hotspot identification, there are some studies on discovering the underlying demand patterns. For example, Lee et al. [18] investigated taxi pickup patterns in Jeju based on telematic data; another work [19] studied the demand and supply behaviors of taxis in Berlin and found that demand can be characterized by recurrent patterns.

Our work is different from these studies on demand analysis. **First**, the demand in these studies is not the real demand since they just consider the orders that have been served. This obviously violates the real application scenario. For example, given a taxi stand with 10 passengers (real demand) and assume that each passenger needs a taxi, the demand is 5 in these studies if there are only five taxis appearing. In contrast, our work will consider all the 10 orders from users. **Second**, they predict the demand while we aim to predict the unmet demand which is affected by both demand and supply. According to Fig. 2, the regularity of unmet demand is much weaker than that of demand, meaning that it is more difficult to predict unmet demand. **Third**, most of these studies predict demand only based on the demand itself while ignoring urban context factors, e.g., weather and traffic. However, as demonstrated by our experiments, these exogenous factors can help to improve the prediction.

B. Unmet Demand Analysis

Unmet demand refers to the number of passengers who fail to find a taxi. In practice, it is difficult to get the unmet demand in traditional taxi services. Therefore, existing studies on unmet demand prediction aim to estimate the degree of unmet demand rather than predicting the volume of unmet demand.

Afian et al. [20] estimated the unmet demand indirectly based on the assumption that more available taxis indicate smaller unmet demand. Shao et al. [21] estimated the demand-supply level of a region by evaluating how long it takes for an available taxi to be occupied after entering the region. Obviously, both [20] and [21] only estimated the degree of unmet demand rather predicting the unmet demand itself. Zhao et al. [22] estimated the unmet demand based on the observation that unmet demand is proportional to the met demand, which is still an indirect approach. In addition, there is also some statistical analysis on the demand-supply relationship. For example, Huang et al. [23] identified the regions of service disequilibrium by using Bayesian spatial scan statistics and Poisson-based hypothesis testing.

Different from traditional taxi services, it is possible to get the real unmet demand in on-demand transport services since all users send order requests to the service center. Therefore, we aim to predict the volume of unmet demand directly, which can make it easier for service companies to know how many drivers they should dispatch to alleviate the impending demand-supply imbalance.

III. Problem Statement

A. Problem Formulation

In on-demand transport services, an order o is denoted by $o{=}(l, t, t_r)$, where l is the spatial location of o, t is the timestamp, and t_r is the response time (i.e., the time for o to get accepted). If order o is not accepted by any drivers, we have $t_r{=}\infty$.

Assuming that a city is partitioned into n disjoint regions $R_1, R_2, \ldots, R_n$ and each day is divided into m time slots $T_1, T_2, \ldots, T_m$, demand is then defined as below.

Definition 1: (***Demand***) Given a region R_i and a time slot T_j, the orders in R_i within time slot T_j is denoted by $D_{i,j}{=}\{o_1, o_2, \ldots\}$, where $\forall o \in D_{i,j}$, $o.l \in R_i$ and $o.t \in T_j$. Then, the number of orders in $D_{i,j}$ is called the demand of region R_i in time slot T_j. □

The unmet orders in region R_i within time slot T_j is denoted by $UD_{i,j}{=}\{o | o \in D_{i,j} \wedge o.t_r{<}\theta\}$, where θ is a time threshold to decide whether an order is responded in time and its default value is 600 seconds. Therefore, one order is regarded as unmet if it is not responded within θ. For simplicity, we record $u_{i,j}{=}|UD_{i,j}|$ where $|UD_{i,j}|$ is the cardinality of $UD_{i,j}$, and call $u_{i,j}$ the unmet demand of region R_i in time slot T_j. Accordingly, the unmet rate $\rho_{i,j}$ is computed by $\rho_{i,j}{=}\frac{u_{i,j}}{|D_{i,j}|}$. For each region R_i, we have a sequence of unmet demand records:

$$S_i = \{u_{i,1}, u_{i,2}, \ldots, u_{i,j}\} \tag{1}$$

We then have the definition of unmet demand prediction problem as below.

Definition 2: (***Unmet demand prediction***) Given a region R_i and its historical unmet demand sequence S_i, an unmet demand prediction problem aims to predict the unmet demand $u_{i,j+1}$ in the next time slot T_{j+1}. □

Essentially, unmet demand prediction problem predicts the number of orders that will not be responded in the next time slot. Intuitively, unmet demand prediction is dependent on the distributions of demand and supply. Moreover, the unmet demand is also time-dependent and affected by many other factors, such as traffic and weather. Therefore, predicting unmet demand requires to combine these factors altogether. To this end, we propose a general framework to extract as many as possible the factors that are relevant to unmet demand. These factors are then integrated to train prediction models.

Table I summarizes the frequently used notations and their meanings in this work.

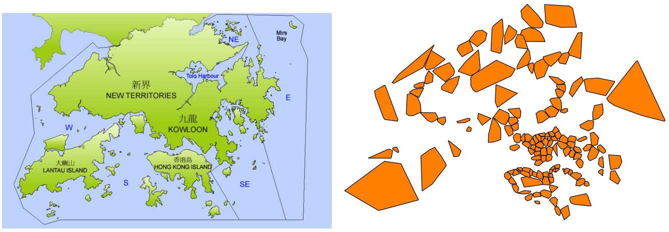
TABLE I
Frequently used notations.

Notation	Meaning
R_i	the i-th region
T_j	the j-th time slot
$o{=}(l, t, t_r)$	an order with location, tiemstamp, and response time
$D_{i,j}$	the orders in region R_i within time slot T_j
$UD_{i,j}$	the unmet orders of region R_i in time slot T_j
$u_{i,j}$	the unmet demand of region R_i in time slot T_j
$\rho_{i,j}$	the unmet rate of region R_i in time slot T_j
θ	order response time threshold
S_i	unmet demand sequence of region R_i
E_i	the entropy of unmet demand sequence S_i
Π^{max}	the maximum predictability of unmet demand in a region

B. Datasets

In this work, we take the order data of GoGoVan in Hong Kong as an example. The order data covers 61 consecutive days in 2016. During this period, we collected about one million orders among which 9% are not responded if the response time threshold θ is set to 600 seconds.

Each day is divided into 24 time slots, i.e., one hour corresponds to one time slot. The whole city of Hong Kong is first partitioned into regions according to its adminstration partition. Then, we narrow down each region's area according to the locations of historical orders and remove those parts without any orders. Finally, as illustrated in Fig. 3, we divide Hong Kong into 144 regions, where Fig. 3(a) is the administration area of Hong Kong[4]. The white space between adjacent regions in Fig. 3(b) could be mountains, forests and water areas that have few orders. Note that, the proposed solution in this study is general and suitable for other region partition methods, e.g., grid partition.



(a) Adminstration area
(b) Partitioned regions

Fig. 3. Hong Kong adminstration area and the partitioned regions.

In addition to *order data*, we also collect *road network data*, *POI data*, *driver trajectory data* and *weather data*. The details of these datasets will be discussed in Section VI.

[4] https://www.afcd.gov.hk

IV. Predictability of Unmet Demand

Before designing solutions for predicting unmet demand, it is important to evaluate the predictability (i.e., regularity) of unmet demand. Generally, high predictability in unmet demand indicates that we can design effective prediction algorithms with high prediction accuracy. In contrast, if the predictability of unmet demand is low, we cannot effectively predict the unmet demand using any prediction algorithms. Therefore, in this section, we explore the predictability of unmet demand using the entropy of unmet demand sequence.

Given an unmet demand sequence S_i of region R_i, the corresponding entropy E_i of S_i is computed by the following formula [24]:

$$E_i = -\sum_{s \subseteq S_i} p(s) \log_2[p(s)] \tag{2}$$

where s represents any subsequence of S_i and $p(s)$ is the probability that s appears in S_i. According to the implication of entropy, a smaller E_i indicates a higher predictability.

In Eq (2), it is computationally prohibitive to enumerate all the subsequences of S_i since the corresponding time complexity is $O(2^{|S_i|})$, where $|S_i|$ is the length of S_i. To deal with this issue, we estimate the entropy E_i by using Lempel-Ziv estimator [25]:

$$\hat{E}_i = \Big(\frac{1}{|S_i|}\sum_{t=1}^{|S_i|} |s_t|\Big)^{-1} \ln |S_i| \tag{3}$$

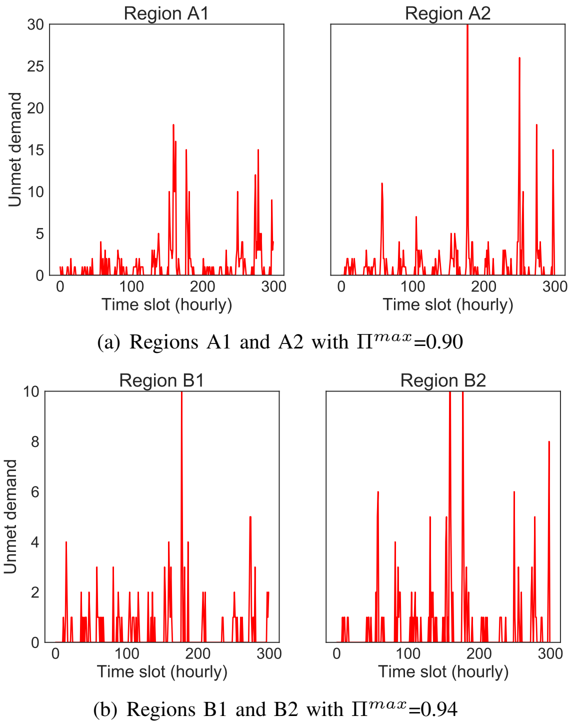
where s_t is the shortest subsequence that starts from the t-th time slot and never appears in previous time slots of S_i, and $| * |$ computes the length of a sequence.

Given the computed entropy $\hat{E}_i$ of S_i and an arbitrary prediction algorithm Ω, the corresponding predictability Π_Ω of Ω should satisfy $\Pi_\Omega \le \Pi^{max}$ according to the Fano's inequality [24], where Π^{max} is the maximum predictability computed by solving the following equation.

$$\hat{E}_i = -\Pi^{max} \log_2(\Pi^{max}) - (1-\Pi^{max}) \log_2(1-\Pi^{max}) + (1-\Pi^{max}) \log_2(\Gamma_i - 1) \tag{4}$$

where $0 \le \Pi^{max} \le 1$, and Γ_i is the number of distinct unmet demand records in region R_i. Maximum predictability Π^{max} means that the prediction accuracy could be as high as Π^{max} if an ideal prediction algorithm is designed.

According to the experiments, the maximum predictability of unmet demand prediction for most regions is larger than 0.9, which indicates that unmet demand is highly predictable. Specifically, Fig. 4 illustrates the unmet demand in regions with different maximum predictabilities.



(a) Regions A1 and A2 with Π^{max}=0.90

(b) Regions B1 and B2 with Π^{max}=0.94

Fig. 4. Unmet demand sequences of regions with different maximum predictabilities.

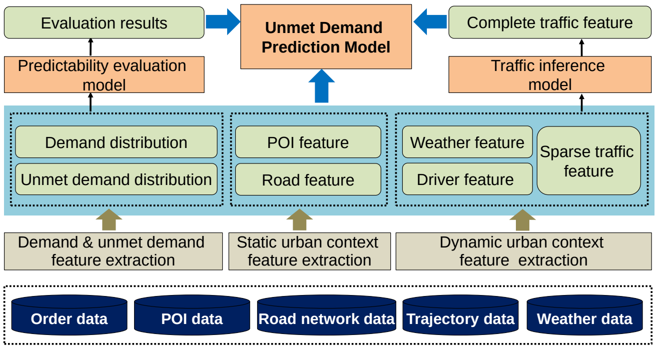


Fig. 5. Solution framework for unmet demand prediction.

V. Solution Framework Overview

Fig. 5 presents the framework proposed for predicting unmet demand in on-demand transport services. At the bottom layer of the framework, we have data from different domains to cover as many as possible the factors that affect unmet demand. Specifically, order data is from logistics domain; POI data about all kinds of urban entities is from commercial domain; weather data is from meteorology domain; and road network data and driver trajectory data are from transportation domain.

First, we extract demand features, including the distributions of demand and unmet demand from the order data and evaluate the predictability of unmet demand. **Second**, we extract static urban features from road network data and POI data to capture the distributions of roads and commercial entities. **Third**, dynamic urban features, such as weather, driver distribution and traffic, are extracted from weather data and trajectory data. Considering that some regions may not have enough trajectories to compute the traffic, we introduce a traffic inference model to deal with the sparsity issue. **Finally**, we fuse all the features extracted from different data sources together and feed them into multiple prediction models to conduct the unmet demand prediction.

The details of these components in Fig. 5 will be elaborated in Sections VI and VII.

VI. Feature Extraction

A. Demand & Unmet Demand Feature Extraction

Generally, the demand in adjacent time slots are highly correlated. Therefore, many demand prediction models utilize time series analysis to achieve the prediction. Similarly, the unmet demand in adjacent time slots are also correlated. Therefore, to predict the unmet demand in the next time slot, the unmet demand records in previous time slots are very important. We thus consider the number of unmet orders (i.e., unmet demand) and the total number of orders (i.e., demand) in previous η time slots, where η is set to 3 in this work.

In addition, for each region, we also consider the total number of orders and the total unmet demand in history, and the average unmet demand of each time slot in history.

B. Static Urban Context Feature Extraction

The unmet demand of a region is highly dependent on the functionalities of that region. For example, commercial regions generally have more orders than residential regions. Considering that the functionalities of a region can be captured by its road network and points of interest (POI), we thus extract road and POI features for each region.

Road feature extraction. The road network of Hong Kong (cf. Fig. 6) is extracted from OpenStreetMap[5] and its total distance is around 1,500 km. The road network has five categories of roads, i.e., *motor way*, *trunk road*, *primary road*, *secondary road* and *tertiary road*. Since each category of roads has the corresponding link roads, there are 10 categories of roads in total. We compute for each region the total length of roads in each category. For example, Fig. 7 illustrates the total length of roads in each category for a selected region.

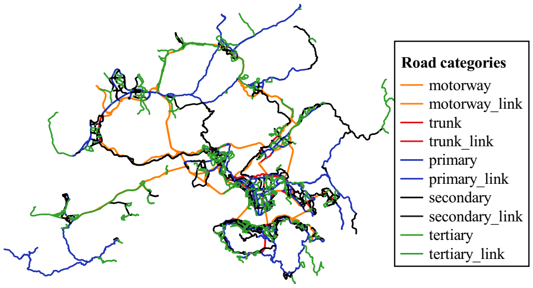


Fig. 6. Hong Kong road network.

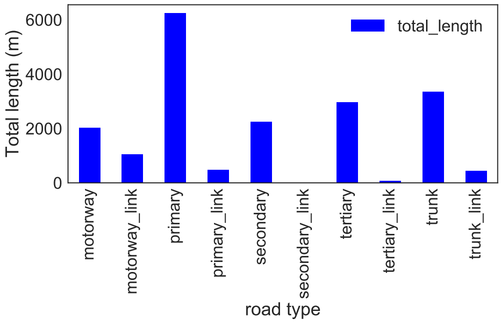


Fig. 7. The distribution of roads in a selected region.

POI feature extraction. We extract around 12,000 POIs from Google Map. These POIs are classified into 84 categories and each POI belongs to at least one category. In this study, we consider the top-10 categories (i.e., *food*, *restaurant*, *transit station*, *bus station*, *store*, *school*, *lodging*, *place of worship*, *health* and *cafe*) which account for about 80% of all POIs. For each region, we compute the number of POIs in each of the top-10 categories. For example, Fig. 8 illustrates the distribution of POIs in a region.

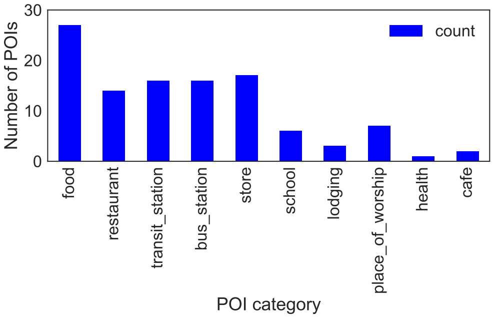


Fig. 8. The distribution of POIs in a region.

C. Dynamic Urban Context Feature Extraction

Dynamic urban context features, such as traffic, weather and driver distribution, are also important for unmet demand prediction because they will affect the behaviors of users and drivers. Therefore, we extract traffic feature, weather feature and driver distribution feature to capture urban dynamics.

Traffic feature extraction. Intuitively, urban traffic affects the volume of unmet demand. On the one hand, bad traffic reduces the working efficiency of drivers since a longer time is required to travel the same route. On the other hand, drivers usually try to avoid going to the regions of bad traffic, thus affecting the orders in these regions or orders with these regions as their destinations.

In this work, the traffic in each region is computed based on drivers' GPS trajectories. A GPS trajectory Tr consists of a sequence of GPS points, i.e., $Tr{=}\langle p_1, p_2, \ldots, p_n\rangle$, where each point $p_i{=}(l, t, s)$ $(1{\le}i{\le}n)$ with l, t and s denoting its spatial location, timestamp, and the current status (idle or busy), respectively. In general, $p_i.l$ records the longitude and latitude of p_i. For a pair of consecutive GPS points (p_i, p_{i+1}), the corresponding average speed between p_i and p_{i+1} is

$$v_i = \frac{dist(p_i.l, p_{i+1}.l)}{p_{i+1}.t - p_i.t} \tag{5}$$

where $dist(p_i.l, p_{i+1}.l)$ computes the road network distance between $p_i.l$ and $p_{i+1}.l$.

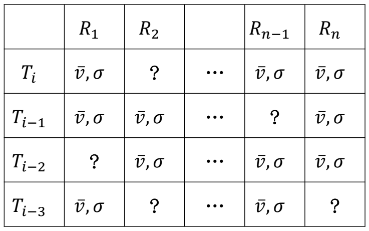
After computing the speeds for all pairs of consecutive GPS points in all driver trajectories, we can obtain a set of speed samples $V_i{=}\{v_1, v_2, \ldots, v_m\}$ for each region R_i by mapping speed samples to the regions according to their spatial locations and the geometric shapes of regions. Then, the traffic in region R_i is described by the average speed $\bar{v}$ and the standard deviation σ of V_i, i.e.,

$$\bar{v} = \frac{\sum_{v\in V_i} v}{m}, \quad \sigma = \sqrt{\frac{\sum_{v\in V_i}(v-\bar{v})^2}{m}}$$

For each region, $\bar{v}$ and σ are computed for each time slot T_j. Therefore, each time slot is associated with a pair ($\bar{v}$, σ).

[5] An online open source map, https://www.openstreetmap.org

Due to the limited number of drivers, we cannot always have enough GPS trajectories to cover all regions in every time slot. Fig. 9 illustrates the traffic information for the current time slot T_i and previous three time slots. We call this matrix traffic matrix and denote it by M_T. Obviously, some regions have no traffic information. To extract traffic features for all regions, we need to infer the missing entries in M_T.



	R_1	R_2		R_{n-1}	R_n
T_i	$\bar{v},\sigma$	?	⋯	$\bar{v},\sigma$	$\bar{v},\sigma$
T_{i-1}	$\bar{v},\sigma$	$\bar{v},\sigma$	⋯	?	$\bar{v},\sigma$
T_{i-2}	?	$\bar{v},\sigma$	⋯	$\bar{v},\sigma$	$\bar{v},\sigma$
T_{i-3}	$\bar{v},\sigma$	?	⋯	$\bar{v},\sigma$	?

Fig. 9. Traffic matrix from time slot T_{i-3} to time slot T_i.

Intuitively, we have two observations. First, the traffic in one region is usually affected by the traffic in its neighbor regions, i.e., neighboring regions have similar traffic in high probability. Second, if the distributions of roads and POIs in two regions are similar, their traffic might also be similar. With these two observations, we leverage context-aware matrix factorization [26] to infer the missing entries in M_T.

First, we generate the urban context feature matrix M_F. For each region R_i, the extracted features in Section VI-B form one vector. Urban context feature matrix M_F is generated by combining the vectors of all regions. Then, we jointly factorize matrices M_T and M_F, i.e.,

$$M_T = X \times Y^T, M_F = Y \times Z^T \tag{6}$$

where X, Y, and Z are latent factors. The corresponding objective function is

$$L(X,Y,Z) = \frac{1}{2}||M_T - X \times Y^T||^2 + \frac{1}{2}\lambda_1 \cdot ||M_F - Y \times Z^T||^2 + \frac{1}{2}\lambda_2 \cdot (||X||^2 + ||Y||^2 + ||Z||^2)$$

We utilize gradient descent method to compute good approximate values for matrices X, Y, and Z, i.e., $\hat{X}$, $\hat{Y}$, $\hat{Z}$. Finally, we have a complete traffic matrix $\hat{M}_T{=}\hat{X} \times \hat{Y}^T$ in which missing entries are filled out.

Weather feature extraction. Unmet demand is also relevant to the weather conditions, especially for taxi services. For example, compared with a sunny day, more citizens would like to take a taxi when it is raining. According to [2], we roughly classify all weather conditions into three categories, i.e., *good weather*, *bad weather* and *very bad weather*. The corresponding weather conditions in each category are detailed in Table II. The weather data is from Weather Underground[6] which updates weather condition every half hour.

Driver distribution feature extraction. Unmet demand also depends the distribution of drivers. Compared with traditional taxi service, on-demand transport services are more flexible since drivers can pick up orders beyond the visual range. However, considering the travel cost, drivers in on-demand transport services still tend to pick up orders near to their current locations. Therefore, to predict unmet demand, we still need to consider the spatial distribution of drivers.

TABLE II
WEATHER CONDITIONS IN EACH CATEGORY.

Category	Weather condition
good weather	partly cloudy, scattered clouds, haze, clear, mostly cloudy
bad weather	light rain showers, light rain, light drizzle, light thunderstorms and rain
very bad weather	rain showers, heavy rain showers, and thunderstorms

For each region R_i, we compute those drivers who have been to R_i in each time slot T_j based on their GPS trajectories and assume that the corresponding set of drivers is $B_{i,j}{=}\{d_1, d_2, \ldots, d_n\}$. For each driver, the GPS points also record the status, i.e., busy with an order or idle. For example, Fig. 10 shows the GPS trajectory of a driver within one hour, where red GPS points mean that the driver is busy while green ones indicate that he is idle.

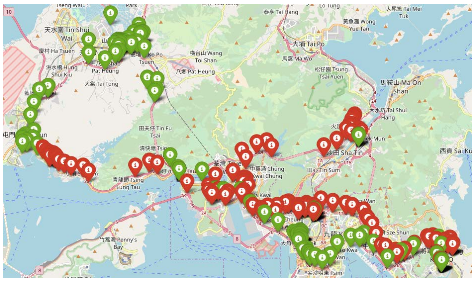


Fig. 10. Driver trajectory where red means busy and green means idle.

For each driver $d{\in}B_{i,j}$, we first compute its busy time span t_1 and idling time span t_2 in region R_i within time slot T_j. After that, for each region R_i, we compute the following features to describe driver distribution.

- The number of drivers appeared in region R_i within time slot T_j, i.e., $|B_{i,j}|$;
- The average idle ratio for all drivers in $B_{i,j}$, i.e., $\frac{1}{|B_{i,j}|}\sum_{d\in B_{i,j}} \frac{t_2}{t_1+t_2}$, and the corresponding standard deviation.

VII. Model Training

Table III summarizes all the extracted features that will be further used for training unmet demand prediction models.

TABLE III
SUMMARY FOR EXTRACTED FEATURES.

Feature type	Features	Count
demand & unmet demand feature	the demand and unmet demand in previous three time slots, average unmet demand, total unmet demand, and total demand	9
road feature	the total length of roads, and lengths of ten categories of roads	11
POI feature	the total number of POIs, and numbers of POIs in the top-10 categories	11
traffic feature	average speed, and speed deviation	2
weather feature	the type of weather (good, bad or very bad)	1
driver distribution feature	the number of drivers, average idle ratio, and idle ratio deviation	3
others	region size, time slot of the day, and day of the week	3

[6] https://www.wunderground.com/

A. Model Description

We introduce three prediction models, i.e., Ridge Regressor (Ridge), Least Absolute Shrinkage and Selection Operator (Lasso) and Random Forest (RF) Regressor, which are briefly described as below.

Ridge Regressor. Ridge regressor improves Ordinary Least Squares by imposing a l_2 penalty on the sizes of coefficients and its objective function is

$$\min_{w} ||Xw - y||_2^2 + \alpha||w||_2^2 \tag{7}$$

where w is the coefficient vector, X is the feature matrix, y is the label vector, $\alpha \geq 0$ is a penalty parameter to control the shrinkage, and $|| * ||_2$ computes the l_2-norm.

Lasso. Lasso [27] performs feature selection and regularization simultaneously. Its general objective function is

$$\min_{w} \frac{1}{2n}||Xw - y||_2^2 + \beta||w||_1 \tag{8}$$

where n is the number of data samples, $|| * ||_1$ computes the l_1-norm, and β is a penalty parameter to adjust the weight of regularization term $||w||_1$.

Random Forest Regressor. RF regressor is an ensemble learning model. Considering that a single decision tree might overfit the training datasets, RF regressor trains multiple decision trees on randomly selected subsets of features and reports the average predicted value of all trees as the final result (cf. Eq. (9)), thus reducing the variance.

$$f(x) = \frac{1}{g}\sum_{i=1}^{g} f_i(x) \tag{9}$$

where $f_i(x)$ is the output of the i-th tree for sample x, g is the number of decision trees, and $f(x)$ is the final result.

Baseline Prediction Model. In addition to the three models above, we also adapt the demand prediction model, Sliding-Window Ensemble Method (SWEM), in [4] to predict unmet demand. SWEM is an ensemble method that integrates three prediction models, i.e., Time-Varying Poisson model, Weighted Time-Varying Poisson model and Autoregressive Integrated Moving Average (ARIMA) model.

B. Data Samples for Training

In the experiments, we generate 144∗24∗61=210,816 samples, where 144, 24 and 61 correspond to the number of regions, the number of time slots in each day and the number of days, respectively. The first 3/4 samples are used for training while the remaining ones are used for testing.

Generally, it is important yet difficult to tune the hyperparameters of regression models in time-dependent applications since these parameters could also be time-dependent. To resolve this issue, we divide time slots of each day into three groups according to the temporal distribution of unmet demand (cf. Fig. 11), where groups D1=$\{0, 1, 2, 3, 4, 5, 6, 20, 21, 22, 23\}$, D2=$\{7, 8, 9, 10, 11\}$ and D3=$\{12, 13, 14, 15, 16, 17, 18, 19\}$. In addition, we use D0 to represent all time slots, i.e., D0=D1∪D2∪D3. Then, we train separate models for these groups.

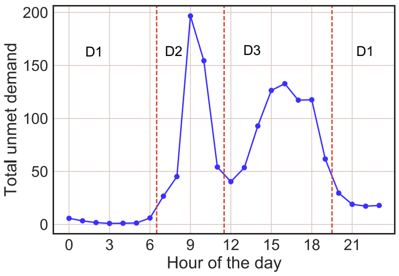


Fig. 11. The temporal distribution of average unmet demand within 61 days.

C. Hyper-parameter Tuning

The hyper-parameters in prediction models are tuned using Grid Search [28] along with cross validation. Concretely, we tune the hyper-parameters in these models as below.

Ridge Regressor. Ridge regressor needs to tune the l_2 penalty parameter α. We vary α from 0.01 to 0.1 with a step of 0.01, from 0.1 to 1.0 with a step of 0.1, and from 1.0 to 10.0 with a step of 1. For each value, we evaluate the performance of Ridge regressor using 10-fold cross-validation.

Lasso. Lasso needs to tune the l_1 penalty parameter β. Fig. 12 illustrates the tuning curves of β on four data groups. According to the illustration, we can get the appropriate value (i.e., the vertical line) for β.

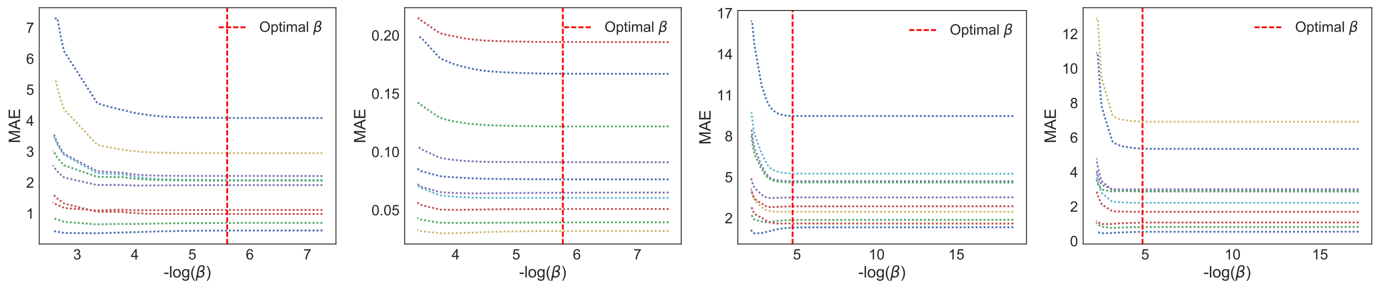
Random Forest Regressor. RF regressor has two hyper-parameters, the number of trees ($ntree$) and the number of features selected in each split node ($mtry$). In Grid Search, we vary $ntree$ from 10 to 200 with a step of 10 and vary $mtry$ from 2 to the total number of features with a step of 1. For each pair of $ntree$ and $mtry$, we use 5-fold cross-validation to evaluate the performance of RF regressor.

SWEM. In SWEM, the parameters in ARIMA are determined by autocorrelation analysis and partial autocorrelation analysis for each region. The size of sliding window H is set to 4 hours. The smoothing factor α' (to distinguish it from the α in Ridge regressor) in Weighted Time-Varying Poisson model is set to 0.4 and the number of historical time slots is set to 8. All the other parameters are set as default (cf. [4]).

The settings of hyper-parameters for all models on different groups of samples are presented in Table IV, where columns **Parameters(A)** and **Parameter(S)** correspond to the models using all features and selected features, respectively. Feature selection will be discussed in next section. Note that SWEM is irrelevant to the features. For simplicity, we put its parameters in the column **Parameters(A)**.

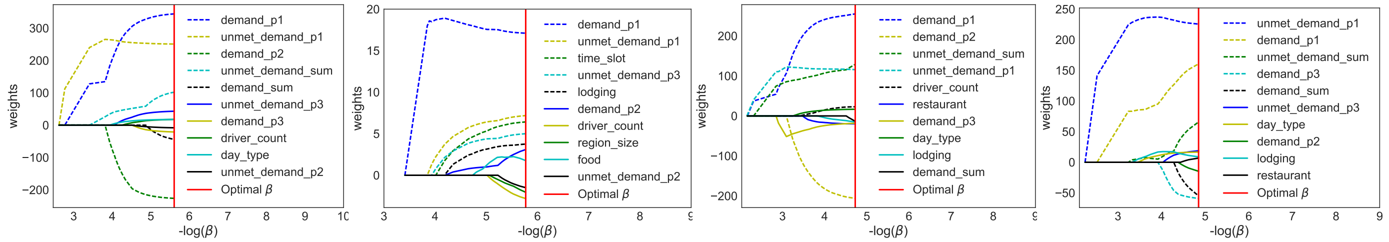
D. Feature Selection

As summarized in Table III, we extract many features that are thought to be relevant to unmet demand. However, some features may have quite limited effect on unmet demand prediction. Therefore, we introduce feature selection to identify the important features for prediction. According to [1], Lasso performs well in selecting important features. Therefore, we use Lasso to conduct feature selection. Fig. 13 illustrates the top-10 most important features for each group of samples, where the vertical line is the optimal β and serves as a cut-off. Those features with values observably larger than 0 at the cut-off are selected.



(a) D0 (b) D1 (c) D2 (d) D3

Fig. 12. Parameter (β) tuning in Lasso.



(a) D0 (b) D1 (c) D2 (d) D3

Fig. 13. Feature selection using Lasso, where the vertical line corresponds to the optimal β.

VIII. Experimental Evaluation

A. Evaluation Metrics

Three evaluation metrics, MAE, RMSE and MAPE, are used for performance evaluation and their meanings are discussed briefly as below.

Mean absolute error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|u_i - \hat{u}_i| \tag{10}$$

where u_i is the real unmet demand, $\hat{u}_i$ is the predicted unmet demand, and N is the number of samples.

Root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(u_i - \hat{u}_i)^2} \tag{11}$$

Mean absolute percent error (MAPE):

$$MAPE = \frac{1}{N}\sum_{i=1}^{n}\frac{|u_i - \hat{u}_i|}{u_i} \tag{12}$$

Considering that u_i could be zero, we compute MAPE with the following revised equation.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|u_i - \hat{u}_i|}{1 + u_i} \tag{13}$$

B. Experimental Results

Predicting unmet demand in time slot T_{j+1}. Table IV presents the results of four prediction models on four groups of samples. MAE-A and MAE-S represent the MAE with all features and selected features, respectively. It is the same for RMSE and MAPE. Since the baseline solution SWEM is irrelevant to the features, we put its results in columns MAPE-A, RMSE-A and MAPE-A for comparison and fill other columns with asterisks. According to Table IV, RF regressor outperforms other models in almost all cases. However, RF regressor performs a little worse on the selected features than on all features. In contrast, both Ridge and Lasso perform better on the selected features. In terms of MAE, SWEM performs better than Ridge and Lasso on D0 and D2, but performs worse than Ridge and Lasso on D1 and D3.

Varying the response time threshold θ. One advantage of the proposed framework is its applicability for different definitions of unmet demand. Intuitively, if an order is not responded in a long time, we can regard it as unmet. Therefore, given different response time thresholds, the unmet demand will vary accordingly. Table V presents the MAE when setting the response time threshold to 300 seconds. Actually, we can even set the response time threshold to zero to predict the total number of orders i.e., demand (cf., Table VI). According to Tables V and VI, RF regressor still outperforms other three models. SWEM also achieves satisfying performance. Therefore, it is reasonable to predict unmet demand and demand using time series data only when the response time threshold is small. However, if we extract more features and feed them into ensemble models like random forest regressor, we can get better prediction results.

TABLE V
MAE when response time threshold θ =300 seconds.

Models	D0	D1	D2	D3
Ridge	0.586	0.112	0.948	0.891
Lasso	0.593	0.113	0.951	0.892
RF	**0.460**	**0.096**	**0.657**	**0.826**
SWEM	0.482	0.105	0.711	0.857

Predicting unmet rate. As discussed in the problem definition, we can also represent the unmet demand in another way, i.e., unmet rate. Therefore, we also evaluate the performance of

TABLE IV
RESULTS FOR PREDICTING UNMET DEMAND IN TIME SLOT T_{i+1}.

Datesets	Models	Parameters (A)	Parameters (S)	MAE-A	MAE-S	RMSE-A	RMSE-S	MAPE-A	MAPE-S
D0	Ridge	$\alpha = 0.1$	$\alpha = 0.06$	0.487	0.470	1.622	1.627	0.251	0.231
	Lasso	$\beta = 2.16E^{-06}$	$\beta = 0$	0.497	0.482	1.611	1.614	0.260	0.243
	RF	mtry=9, ntree=150	mtry=5, ntree=130	**0.398**	**0.439**	**1.408**	**1.547**	**0.186**	**0.205**
	SWEM	$\alpha' = 0.4, \gamma = 8, H = 4$	$*$	0.438	$*$	1.641	$*$	0.193	$*$
D1	Ridge	$\alpha = 0.1$	$\alpha = 0.06$	0.081	0.095	0.382	0.391	0.051	0.063
	Lasso	$\beta = 1.70E^{-06}$	$\beta = 0$	0.082	0.096	0.383	0.391	0.052	0.064
	RF	mtry=9, ntree=130	mtry=5, ntree=120	**0.071**	**0.085**	**0.369**	**0.385**	**0.044**	**0.054**
	SWEM	$\alpha' = 0.4, \gamma = 8, H = 4$	$*$	0.114	$*$	0.442	$*$	0.070	$*$
D2	Ridge	$\alpha = 0.1$	$\alpha = 0.06$	0.802	0.764	2.337	2.340	0.392	0.366
	Lasso	$\beta = 1.88E^{-05}$	$\beta = 0$	0.825	0.775	2.308	2.307	0.415	0.379
	RF	mtry=7, ntree=120	mtry=4, ntree=120	**0.594**	**0.618**	**1.791**	**1.997**	0.259	0.257
	SWEM	$\alpha' = 0.4, \gamma = 8, H = 4$	$*$	0.624	$*$	1.923	$*$	**0.253**	$*$
D3	Ridge	$\alpha = 0.2$	$\alpha = 0.08$	0.756	0.743	1.926	**1.917**	0.368	**0.346**
	Lasso	$\beta = 1.42E^{-05}$	$\beta = 0$	0.762	**0.740**	1.918	1.927	0.371	0.348
	RF	mtry=7, ntree=120	mtry=4, ntree=120	**0.706**	0.757	**1.901**	2.004	**0.321**	**0.346**
	SWEM	$\alpha' = 0.4, \gamma = 8, H = 4$	$*$	0.764	$*$	1.952	$*$	0.340	$*$

TABLE VI
MAE WHEN RESPONSE TIME THRESHOLD θ =0 SECOND.

Models	D0	D1	D2	D3
Ridge	2.193	0.487	3.14	2.901
Lasso	2.129	0.477	3.029	2.849
RF	**1.182**	**0.322**	**1.599**	**2.089**
SWEM	1.403	0.476	1.912	2.332

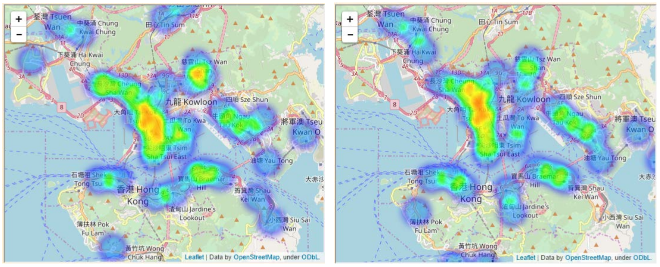
proposed methods for predicting unmet rate. As illustrated by Table VII, the RF regressor still achieves better performance than the baseline method SWEM.

TABLE VII
MAE WHEN PREDICTING UNMET RATE, WHERE RESPONSE TIME THRESHOLD θ =600 SECONDS.

Models	D0	D1	D2	D3
Ridge	0.0693	0.0415	0.0857	0.0908
Lasso	0.0693	0.0414	0.0856	0.0908
RF	**0.0647**	**0.0412**	**0.0804**	**0.0867**
SWEM	0.0783	0.0669	0.0866	0.0903

C. Case Study

Fig. 14 illustrates the heatmaps of predicted unmet demand using RF regressor and real unmet demand at 9am. As suggested by the illustration, the predicted results match the real unmet demand well.



(a) Predicted unmet demand

(b) Real unmet demand

Fig. 14. Predicted unmet demand and real unmet demand.

IX. Conclusion

In this study, we propose a general framework for predicting unmet demand in on-demand transport services. Under the framework, we first extract both static and dynamic urban features relevant to unmet demand from data in multiple domains. Then, multiple prediction models are trained to predict the unmet demand using the extracted features. As demonstrated via experiments, the proposed framework can effectively predict the unmet demand in the next time slot with different settings of response time threshold.

In future, we would like to study individual drivers' response patterns to orders and analyze the correlation between unmet orders and these patterns. By doing this, we wish to obtain the insights into demand-supply imbalance from a new view.

References

[1] S. M. Hassan, L. Moreira-Matias, J. Khiari, and O. Cats, "Feature selection issues in long-term travel time prediction," in *Advances in Intelligent Data Analysis*, 2016, pp. 98–109.

[2] J. Guan, W. Wang, W. Li, and S. Zhou, "A unified framework for predicting kpis of on-demand transport services," *IEEE Access*, 2018.

[3] N. Davis, G. Raina, and K. P. Jagannathan, "A multi-level clustering approach for forecasting taxi travel demand," in *IEEE International Conference on Intelligent Transportation Systems, ITSC*, 2016, pp. 223–228.

[4] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi-passenger demand using streaming data," *IEEE Trans. Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 2013.

[5] L. Moreira-Matias, J. Gama, M. Ferreira, and L. Damas, "A predictive model for the passenger demand on a taxi network," in *IEEE International Conference on Intelligent Transportation Systems, ITSC*, 2012, pp. 1014–1019.

[6] X. Li, G. Pan, Z. Wu, G. Qi, S. Li, D. Zhang, W. Zhang, and Z. Wang, "Prediction of urban human mobility using large-scale taxi traces and its applications," *Frontiers of Computer Science in China*, vol. 6, no. 1, pp. 111–121, 2012.

[7] D. Zhang, T. He, S. Lin, S. Munir, and J. A. Stankovic, "Taxi-passenger-demand modeling based on big data from a roving sensor network," *IEEE Trans. Big Data*, vol. 3, no. 3, pp. 362–374, 2016.

[8] N. Mukai and N. Yoden, "Taxi demand forecasting based on taxi probe data by neural network," in *International Conference on Intelligent Interactive Multimedia Systems and Services (IIMSS)*, 2012, pp. 589–597.

[9] J. Xu, R. Rahmatizadeh, and L. Boloni, "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Trans. Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–10, 2017.

[10] K. Zhao, D. Khryashchev, J. Freire, C. Silva, and H. Vo, "Predicting taxi demand at high spatial resolution: Approaching the limit of predictability," in *IEEE International Conference on Big Data*, 2016.

[11] B. Jäger, M. Wittmann, and M. Lienkamp, "Analyzing and modeling a citys spatiotemporal taxi supply and demand: A case study for munich," *Journal of Traffic and Logistics Engineering*, vol. 4, no. 2, pp. 147–153, 2016.

[12] G. Pan, G. Qi, Z. Wu, D. Zhang, and S. Li, "Land-use classification using taxi GPS traces," *IEEE Trans. Intelligent Transportation Systems*, vol. 14, no. 1, pp. 113–123, 2013.

[13] D. Liu, S. Cheng, and Y. Yang, "Density peaks clustering approach for discovering demand hot spots in city-scale taxi fleet dataset," in *IEEE International Conference on Intelligent Transportation Systems, ITSC*, 2015, pp. 1831–1836.

[14] L. Zhang, C. Chen, Y. Wang, and X. Guan, "Exploiting taxi demand hotspots based on vehicular big data analytics," in *IEEE 84th Vehicular Technology Conference, VTC*, 2016, pp. 1–5.

[15] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[16] H. Chang, Y. Tai, and J. Y. Hsu, "Context-aware taxi demand hotspots prediction," *IJBIDM*, vol. 5, no. 1, pp. 3–18, 2010.

[17] K. Zhang, Z. Feng, S. Chen, K. Huang, and G. Wang, "A framework for passengers demand prediction and recommendation," in *IEEE International Conference on Services Computing, SCC*, 2016, pp. 340–347.

[18] J. Lee, I. Shin, and G. Park, "Analysis of the passenger pick-up pattern for taxi location recommendation," in *International Conference on Networked Computing and Advanced Information Management*, 2008, pp. 199–204.

[19] J. Bischoff, M. Maciejewski, and A. Sohr, "Analysis of berlin's taxi services by exploring GPS traces," in *International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2015, pp. 209–215.

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

[21] D. Shao, W. Wu, S. Xiang, and Y. Lu, "Estimating taxi demand-supply level using taxi trajectory data stream," in *ICDMW*, 2015, pp. 407–413.

[22] K. Zhao, X. Zheng, and H. T. Vo, "Inferring unmet human mobility demand with multi-source urban data," in *Web and Big Data - APWeb-WAIM International Workshops: MWDA, HotSpatial, GDMA, DDC, SDMA, MASS, 2017, Revised Selected Papers*, 2017, pp. 118–127.

[23] Y. Huang and J. W. Powell, "Detecting regions of disequilibrium in taxi services under uncertainty," in *SIGSPATIAL*, 2012, pp. 139–148.

[24] C. Song, Z. Qu, N. Blumm, and A. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[25] I. Kontoyiannis, P. H. Algoet, Y. M. Suhov, and A. J. Wyner, "Nonparametric entropy estimation for stationary processesand random fields, with applications to english text," *IEEE Trans. Information Theory*, vol. 44, no. 3, pp. 1319–1327, 1998.

[26] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM TIST*, vol. 5, no. 3, pp. 38:1–38:55, 2014.

[27] R. Tibshirani, "Regression shrinkage and selection via the lasso: a retrospective," *Journal of the Royal Statistical Society B*, vol. 73, no. 3, pp. 273–282, 2011.

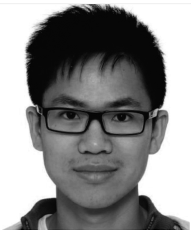
[28] J. Mendes-Moreira, A. M. Jorge, J. F. de Sousa, and C. Soares, "Comparing state-of-the-art regression methods for long term travel time prediction," *Intell. Data Anal.*, vol. 16, no. 3, pp. 427–449, 2012.

Wengen Li received the B.Eng. degree and Ph.D. degree in computer science from Tongji University, Shanghai, China, in 2011 and 2017, respectively. In addition, he received a dual Ph.D. degree in computer science from the Hong Kong Polytechnic University in 2018. He is currently a Postdoctoral Fellow of the Department of Computing at the Hong Kong Polytechnic University. His research interests include spatial data management, and big data analytics for human mobility and urban logistics. He is a member of China Computer Federation (CCF).

Jiannong Cao received the B.Sc. degree in computer science from Nanjing University, China, in 1982, and the M.Sc. and Ph.D. degrees in computer science from Washington State University, USA, in 1986 and 1990 respectively. He is currently a Chair Professor of Department of Computing at The Hong Kong Polytechnic University, Hong Kong. His research interests include parallel and distributed computing, wireless networks and mobile computing, big data and cloud computing, pervasive computing, and fault tolerant computing. He has co-authored 5 books in Mobile Computing and Wireless Sensor Networks, co-edited 9 books, and published over 500 papers in major international journals and conference proceedings. He is a fellow of IEEE, a distinguished member of ACM, a senior member of China Computer Federation (CCF).

Jihong Guan received the bachelor's degree from Huazhong Normal University in 1991, the master's degree from Wuhan Technical University of Surveying and Mapping (merged into Wuhan University since 2000) in 1991, and the PhD degree from Wuhan University in 2002. She is currently a professor in the Department of Computer Science and Technology, Tongji University, Shanghai, China. Before joining Tongji University, she served in the Department of Computer, Wuhan Technical University of Surveying and Mapping from 1991 to 1997, as an assistant professor and an associate professor (since August 2000), respectively. She was an associate professor (2000-2003) and a professor (Since 2003) in the School of Computer, Wuhan University. Her research interests include databases, data mining, distributed computing, bioinformatics, and geographic information systems (GIS).

Shuigeng Zhou received the bachelor's degree from Huazhong University of Science and Technology (HUST) in 1988, the master's degree from the University of Electronic Science and Technology of China (UESTC) in 1991, and the PhD degree in computer science from Fudan University, Shanghai, China, in 2000. He is currently a professor in the School of Computer Science, Fudan University. He served in Shanghai Academy of Spaceflight Technology from 1991 to 1997, as an engineer and a senior engineer (since 1995), respectively. He was a postdoctoral researcher in the State Key Lab of Software Engineering, Wuhan University from 2000 to 2002. His research interests include data management, data mining and bioinformatics.



Guanqing Liang received the B.Sc. degree in telecommunication engineering from Sun Yat-sen University, China, in 2011. He received the Ph.D. degree in the Department of Computing at the Hong Kong Polytechnic University at 2016. He is currently a researcher at Wisers AI Lab, Hong Kong. His research interests include natural language processing, big data analysis and machine learning.

Winnie K.Y. So received the Mphil degree in electronic engineering from City University of Hong Kong in 2017. She worked in Laboratory for Computational Neuroscience and focused on Bio-signal processing and Brain Computer Interface. She joined GoGoVan as a Data Scientist for a year. Now, she is Data Scientist in AXA.

Michal Szczecinski received the Master degree in IT and Econometrics from University of Szczecin in Poland. He is Head of Analytics at GoGoVan. His interest is in building strong analytics capability and using data science to contribute value to the business. Currently he focuses on optimization projects in logistics industry. Previously, he has worked in top professional services, technology and mobile gaming companies.