# *Dmodel*: Online Taxicab Demand Model from Big Sensor Data in a Roving Sensor Network

Desheng Zhang and Tian He
University of Minnesota, USA
{zhang,tianhe}@cs.umn.edu

Shan Lin
Temple University, USA
shan.lin@temple.edu

Sirajum Munir and John A. Stankovic
University of Virginia, USA
{munir,stankovic}@cs.virginia.edu

***Abstract*—Investigating passenger demand is essential for the taxicab business. Existing solutions are typically based on offline data collected by manual investigations, which are often dated and inaccurate for real-time analysis. To address this issue, we propose $Dmodel$, employing roving taxicabs as *real-time mobile sensors* to (i) infer passenger arriving moments by interactions of vacant taxicabs, and then (ii) infer passenger demand by a customized online training with both historical and real-time data. Such huge taxicab data (almost 1TB per year) pose a big data challenge to us. To address this challenge, $Dmodel$ employs a novel parameter called pickup pattern (accounts for various real world logical information, *e.g.*, bad weather) to reduce the size of data to be processed. We evaluate $Dmodel$ with a real world $450$ GB dataset of $14,000$ taxicabs for a half year, and results show that compared to ground truth, $Dmodel$ achieves a $76\%$ accuracy on the demand inference and outperforms a statistical model by $39\%$. We further present an application where $Dmodel$ is used to dispatch vacant taxicabs to achieve an equilibrium between passenger demand and taxicab supply across urban regions.**

## I. Introduction

Understanding and predicting passenger demand are essential for the taxicab business [1]. With accurate knowledge of demand, taxicab companies can schedule their fleet and dispatch individual taxicabs to minimize idle driving time and maximize profit. Historically, such demand has been investigated by manual procedures (*e.g.*, creating surveys [2]). However, these manual studies are often dated, incomplete, and difficult to use in real-time. In particular, passenger demand experiences irregular spatio-temporal dynamics due to real-world phenomena, *e.g.*, bad weather or special events.

To create an accurate demand model we provide a two part solution. First, we mine a large dataset of historical information regarding passenger demand and taxicabs trips. This results in the basis of our method, from which we identify what aspects should be used to infer specific real-time demand. In this work, the historical dataset used is from 14,000 taxicabs for 6 months (450 GB) in a Chinese city, Shenzhen. While this historical model is more accurate than sample-based surveys, it cannot handle many real-time issues and thus has major limitations if used alone.

Second, to address the short term, real-time dynamics, we consider the thousands of roving taxicabs as real-time mobile sensors and collect current information from them. This is possible because taxicabs in dense urban areas are equipped with GPS as location sensors and fare meters as passenger sensors, and thus their locations and occupancy status can be periodically uploaded to a dispatching center. These taxicabs form a real-time "roving sensor network". The streaming data used are from a live data feed in the Shenzhen taxicab network with an average rate of 450 taxicab status records per second.

Admittedly, several systems have proposed to use taxicab GPS traces to infer passenger demand, but they typically have two simplifying assumptions [3] [4] [5]: (i) they assume that the previous demand is given by the picked up passengers, but overlook the waiting passengers who did not get picked up; and (ii) they assume that the current demand can be inferred by the long term historical demand, but overlook the fact that the passenger demand is highly dynamic. For example, after a major concert, due to the high demand, there are few picked up passengers yet numerous waiting passengers, and further the average historical demand cannot indicate the suddenly increased demand due to the concert.

In this paper, to improve these simplifying assumptions, we propose $Dmodel$, which observes online hidden contexts to infer passenger demand based on both historical and real-time data. The contributions of this paper are as follows.

- We identify passenger demand with a combined offline big data analysis and a real-time roving sensor network, where taxicabs detect passenger counts and arriving moments. It is important to note that passenger arriving moments are, in general, unknown. However, a major contribution of $Dmodel$ is how the roving sensor network infers them by utilizing taxicabs' interactions.
- We present a novel parameter, called a *pickup pattern*, to quantify taxicab operational similarity (*i.e.*, how soon a taxicab can pick up a passenger after entering a road segment at a time slot) among different days using big data, *e.g.*, 450 GB in Shenzhen. We show that naively using more data from such a big dataset results in not only an unnecessarily large workload, but also inaccurate inferences. A key novelty of $Dmodel$ is to *utilize the real-time pickup pattern to select customized yet compact training data to increase inference accuracy*. This pickup pattern implicitly accounts for spatio-temporal dynamics in real-world phenomena, *e.g.*, bad weather.
- We test $Dmodel$ on a $450$ GB dataset created by $6$ months of status records from $14,000$ taxicabs in Shenzhen. The evaluations show that compared to ground truth, $Dmodel$ achieves a $76\%$ inference accuracy of demand in terms of the passenger counts. and outperforms a statistical model by $39\%$.
- We show $Dmodel$'s practical value in a real world application where the demand inferred by $Dmodel$ is used to dispatch vacant taxicabs across different regions of a city to achieve a better equilibrium between passenger demand and taxicab supply, which leads to less idle driving time for drivers and waiting time for passengers.

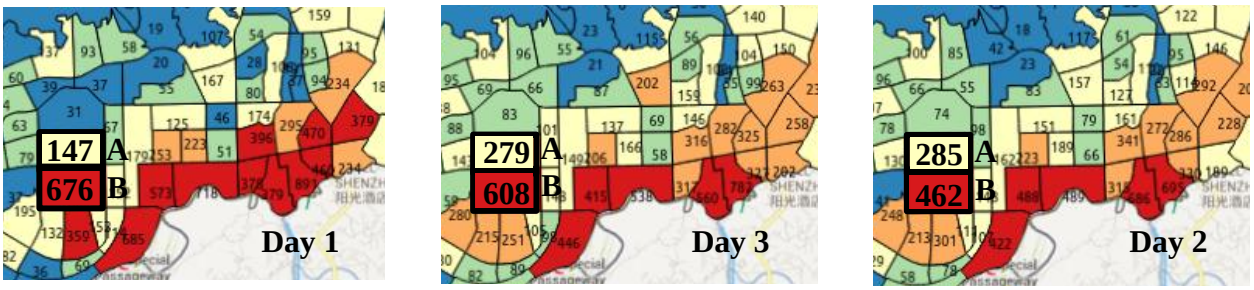


Fig 2: Demand Dynamics for the Same Hourly Slot in Three Different Days

The organization of the paper is as follows. Section II shows motivations. Section III introduces the roving sensor networks. Sections IV and V describe $Dmodel$'s design and evaluation. Section VI proposes the application, followed by the related work and the conclusion in Sections VII and VIII.

## II. Motivations

In this section, based on empirical data (introduced in Section III) from a real-world taxicab network with 14,000 taxicabs in Shenzhen, we present our motivations to improve upon two legacy assumptions for passenger demand analysis.

### A. Assumption on Inference for Previous Demand

**Legacy Assumption One:** Given a previous time slot, the passenger demand (*i.e.*, total counts of all passengers requiring taxicab services) equals to the number of picked up passengers (*i.e.*, pickup counts) [3] [4] [5].

In this work, we argue that though all passengers get picked up eventually, for a previous time slot the passenger demand should include not only the picked up passengers, but also the waiting passengers who had arrived, but did not get picked up. Thus, the passenger demand should equal the total passenger count (*i.e.*, the pickup passenger count plus the waiting passenger count), instead of the pickup passenger count alone. Figure 1 gives the difference between pickup counts (*i.e.*, the demand in the legacy assumption) and total passenger counts (*i.e.*, the demand in our improved assumption) for Shenzhen area in 5 minute time slots.

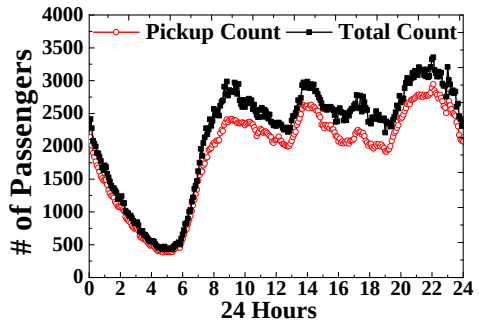


Fig 1: Pickup and Total Counts

We found that the pickup and total passenger counts are usually different especially in the slots during rush hours. Note that a passenger who arrived at one slot and got picked up at the next slot is individually counted in the two slots, *i.e.*, counted as a *waiting passenger* in the first slot, and counted as a *pickup passenger* in the second slot.

The key reason for this legacy assumption is that arriving moments for picked up passengers cannot be obtained by existing infrastructures. Thus, the waiting passenger count of a time slot is unknown. To address this issue, in this paper, we present a novel method based on the interactions of vacant taxicabs to infer the arriving moments, which are used to obtain the waiting passenger counts for more accurate demand analysis. The details are given in Section III-B2.

### B. Assumption on Inference for Current Demand

**Legacy Assumption Two:** Given a current time slot, the passenger demand can be accurately inferred by the historical average passenger demand for the same slot [5].

For this assumption, we argue that for the same area and time slot, the passenger demand experiences irregular temporal dynamics on different days due to various real-world factors, and cannot be accurately inferred without considering more contexts. Figure 2 gives the passenger demand for the same hourly slot in three different weekdays, which is shown by total passenger counts in different administrative regions of Shenzhen. Suppose we want infer the passenger demand of Region $A$ and $B$ on Day 3 given in the middle figure, and the historical demand for the same two regions and the same time slot on Days 1 and 2 as given by the left and right figures. If we infer Region $A$'s demand on Day 3 based on the Region $A$'s demand on Day 1, we only have a $\frac{279-|147-279|}{279} \approx 53\%$ accuracy; similarly, if we infer Region $B$'s demand on Day 3 based on the Region $B$'s demand on Day 2, we only have a $\frac{608-|462-608|}{608} \approx 76\%$ accuracy. Thus, this legacy assumption leads to an inaccurate inference.

The key reason for this assumption in the past is lacking an effective parameter to select the related historical data as training data for the inference. Thus, in this paper, to improve upon this assumption, we propose a novel parameter called the *pickup pattern* to select a customized training dataset for a particular demand inference. For example, based on the pickup pattern, if we find that Region $A$'s demand on Day 2 is more related to Region $A$'s demand on Day 3, then we can infer Region $A$'s demand on Day 3 based on Region $A$'s demand on Day 2. As a result, we can improve the accuracy for Region $A$ on Day 3 from $\frac{279-|147-279|}{279} \approx 53\%$ to $\frac{279-|285-279|}{279} \approx 98\%$.

Note that the above example only gives a straightforward intuition, and the real inference is more complicated. But we found that finding highly related data (instead of all historical data) for the inference can significantly increase the inference accuracy, and reduce the workload by not having to process the entire taxicab dataset. The details are given in Section IV-B1.

## III. The Roving Sensor Network

Recently, taxicab infrastructures in large cities are upgraded with onboard GPS and communication devices and dispatch centers [6]. Built on such infrastructures, a *roving sensor network* consists of (i) numerous *roving taxicabs* in the frontend as mobile sensors to detect passengers, and (ii) a *dispatching center* in the backend to receive sensing records (*i.e.*, taxicab status) to analyze passenger demand. Figure 3 gives a dataset of such sensing records from Shenzhen, the most crowded city in China [7] ($17,150$ people per square KM). This half-year dataset contains almost 4 billion sensing records with a total size of 450 GB.

| Sensing Dataset Summary | |
|---|---|
| Collection Period | 6 Months |
| Collection Date | 01/01-06/30 |
| # of Taxicabs | 14,453 |
| # of Pickup Events | 98,472,628 |
| # of Sensing Records | 3.9 Billion |
| Data Size | 450 GB |

Fig 3: Dataset Summary

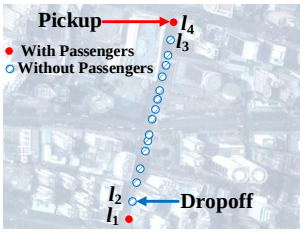


Fig 5: Cruising Events

### A. Detected Events

Based on sensing records, we observe two kinds of events related to passenger demand by tracking taxicabs' status.

*1) Pickup Event:* For the same taxicab, if its status turns from "unoccupied" to "occupied" in two consecutive records, then it indicates that this taxicab just *picked up* a passenger in the location indicated by corresponding GPS signal, which is associated to a *pickup event*; similarly, a *dropoff event* can also be detected. Figure 4 gives a graph representing the corresponding pickup and dropoff events in 245 urban regions in Shenzhen (including the airport, train stations, residential areas, *etc*) from 7AM to 9AM of a Monday. The size of a vertex indicates the number of events in the region; the color of a vertex indicates one of six districts. We remove links with trips fewer than 30 to make the figure clear.

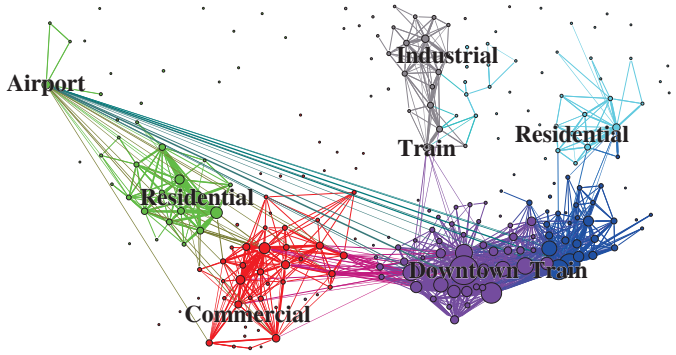


Fig 4: Corresponding Pickup and Dropoff Events

*2) Cruising Event:* A cruising event begins with a dropoff event and ends with a pickup event. Figure 5 gives a cruising event where a taxicab first drops off a passenger between $l_1$ and $l_2$, and then cruises from $l_2$ to $l_3$, and finally picks up a new passenger between $l_3$ and $l_4$. By this cruising event, we infer an absence of passengers on segment from $l_2$ to $l_3$ during the time when this taxi cruises this segment.

### B. Inferred Phenomena

We study two phenomena inferred by the above two events.

*1) Passengers in a Particular Temporal and Spatial Area:* Phenomenon 1 in Figure 6 gives a pickup event $p_i$ where a vacant taxicab $T_i$ cruised a segment $s_j$ and picked up one passenger $P_i$. Based on this observation, we infer that there is only one passenger (*i.e.*, $P_i$) in the dashed temporal and spatial area. This is because if there is another passenger $P_j$, $T_i$ would pick $P_j$ up, which contradicts to the fact that $T_i$ picked up $P_i$ in the pickup event $p_i$. Phenomenon 2 in Figure 6 shows a cruising event where a vacant taxicab $T_j$ cruised a segment $s_j$ and did not pick up any passenger. Based on this observation, we infer that there is no passenger in the dashed temporal and spatial area. This is because if there is a passenger $P_j$, $T_j$ would pick $P_j$ up, which contradicts the fact that $T_j$ did not pick up passengers when it cruised $s_j$. Note that there may be passengers outside the dashed area yet inside the rectangle, since passengers (*e.g.*, $P_k$) can arrive at a location on segment $s_j$, after vacant taxicabs passed this location.

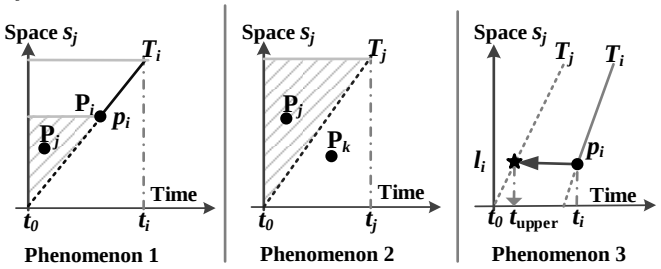


Fig 6: Inferred Phenomena

*2) Arriving Moments of Picked Up Passengers:* An arriving moment indicates the time when a passenger starts to wait for a taxicab, which can be used to obtain the ground truth of the total passenger counts for a segment during a time slot. Accurately obtaining such arriving moments is almost impossible under current infrastructures. But we present a method to obtain the upper bound of an arriving moment $t_{\text{upper}}$, *i.e.*, the earliest possible moment of a passenger starting to wait for a taxicab. As in Phenomenon 3 of Figure 6, assuming passengers do not move significantly when waiting for taxicabs, given a pickup event $p_i$ in terms of pickup moment $t_i$ and location $l_i$, we find the latest cruising event where another vacant taxicab $T_j$ passed the same location $l_i$ (shown as the star). Thus, the moment $t_{\text{upper}}$ when $T_j$ passed $l_i$ is the upper bound of the arriving moment of the passenger $P_i$ in the pickup event $p_i$. This is because if the moment that $P_i$ starts to wait for a taxicab is earlier than this bound $t_{\text{upper}}$, then $P_i$ would be picked up by $T_j$ at $t_{\text{upper}}$, which contradicts the fact that $P_i$ was picked up by $T_i$ at $t_i$. We use this upper bound as the arrival moment, which leads to a lower bound of the arrival count in the latest slot, enabling a conservative inference. Note that waiting passengers' arriving moments cannot be inferred until they are picked up.

Inferred by such a sensor network, the above phenomena provide abundant information with high resolution, used by $Dmodel$ to infer the passenger demand shown as follows.

## IV. *Dmodel* DESIGN

The *Dmodel* is a dynamic inference model for generic passenger demand at road segment levels on a hourly basis. Conceptually, for a segment $s_j$, at the end of a time slot $\tau_i$, *Dmodel* takes both real-time data uploaded in $\tau_i$ and historical data uploaded before $\tau_i$ as input, and produces inferred demand in terms of total passenger count for the next slot $\tau_{i+1}$, by summing up two kinds of passengers as follows.

**Previous Left-behind Passengers** who had arrived at segment $s_j$ before the end of $\tau_i$, and yet were not picked up in $\tau_i$. To obtain this count, *Dmodel* first aggregates real-time pickup events to obtain the *pickup count* for picked up passengers in $\tau_i$. Next, *Dmodel* employs a novel parameter called pickup pattern to obtain customized training data to infer the *total passenger count* (either picked up or not) in $\tau_i$ by corresponding pickup counts. Finally, *Dmodel* obtains the left-behind passenger count by subtracting the pickup count of $\tau_i$ from the total passenger count of $\tau_i$.

**Future Arriving Passengers** who have not arrived yet, but will arrive during $\tau_{i+1}$ at segment $s_j$, *i.e.*, future arrivals. *Dmodel* infers the future arrivals by maintaining a probability distribution of a passenger arrival rate for every road segment. At the end of a time slot $\tau_i$, based on the pickup count in $\tau_i$, *Dmodel* first infers the corresponding passenger arrival in $\tau_i$, and then updates the distribution of arrival rates accordingly, and finally infers the future arrival by this updated distribution.

In what follows, we first present demand modeling, and then elaborate how to obtain these two kinds of passengers.

### A. Passenger Demand Modeling

Key notations for a slot $\tau_i$ and a segment $s_j$ are as follows.

- $\mathbb{P}_{\tau_i}^{s_j}$***: Pickup Count***: The total number of picked up passengers during $\tau_i$ at $s_j$.
- $\mathbb{L}_{\tau_i}^{s_j}$***: Left-behind Count***: The total number of waiting yet not picked up passengers during $\tau_i$ at $s_j$.
- $\mathbb{A}_{\tau_i}^{s_j}$***: Arrival Count***: The total number of arriving passengers during $\tau_i$ at $s_j$.
- $\mathbb{T}_{\tau_i}^{s_j}$***: Total Passenger Count***: The total number of passengers who wait for taxicabs during $\tau_i$ at $s_j$.

In the following, we omit all same superscripts for a concise notation. Figure 7 shows examples of the notation. The $x$-axis is the time, and the $y$-axis is the space, *i.e.*, segment $s_j$. A total of three passengers are picked up, indicated by three pickup events $p_1$, $p_2$ and $p_3$, and the arriving events when they start to wait for taxicabs are given by $a_1$, $a_2$ and $a_3$. As a result, for the time slot $\tau_0$, $\mathbb{A}_{\tau_0}=1$, $\mathbb{P}_{\tau_0}=0$, $\mathbb{L}_{\tau_0}=1$, $\mathbb{T}_{\tau_0}=1$; for the time slot $\tau_1$, $\mathbb{A}_{\tau_1}=2$, $\mathbb{P}_{\tau_1}=2$, $\mathbb{L}_{\tau_1}=1$, $\mathbb{T}_{\tau_1}=3$; for the time slot $\tau_2$, $\mathbb{A}_{\tau_2}=0$, $\mathbb{P}_{\tau_2}=1$, $\mathbb{L}_{\tau_2}=0$, $\mathbb{T}_{\tau_2}=1$.

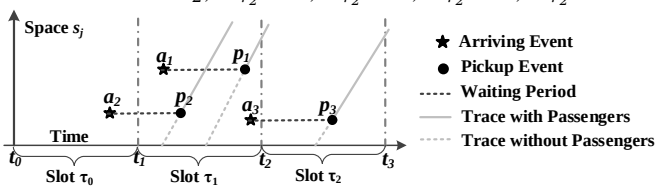


Fig 7: Notation Example

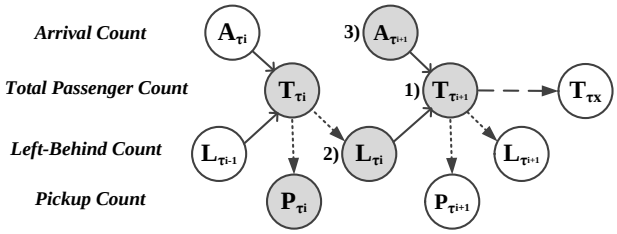


Fig 8: Passenger Demand in a Hidden Markov Model

*1) Demand Modeling:* In Figure 8, we analyze passenger demand as an unobservable state in a Hidden Markov Model.

1) At the end of a time slot $\tau_i$, the key system state that needs to be inferred is the total passenger count $\mathbb{T}_{\tau_{i+1}}$ of the next slot $\tau_{i+1}$, which takes the left-behind count $\mathbb{L}_{\tau_i}$ of $\tau_i$ and the arrival count $\mathbb{A}_{\tau_{i+1}}$ of $\tau_{i+1}$ as two inputs (shown by arrows with solid lines). Thus, we have
$$\mathbb{T}_{\tau_{i+1}} = \mathbb{L}_{\tau_i} + \mathbb{A}_{\tau_{i+1}}.$$
2) As one input for $\mathbb{T}_{\tau_{i+1}}$, the left-behind count $\mathbb{L}_{\tau_i}$ of $\tau_i$ is also one of two outputs (shown by arrows with dashed lines) of the previous system state, *i.e.*, the total passenger count $\mathbb{T}_{\tau_i}$ of $\tau_i$. The other output of $\mathbb{T}_{\tau_i}$ is the observable pickup count $\mathbb{P}_{\tau_i}$ of $\tau_i$. Thus, we have
$$\mathbb{L}_{\tau_i} = \mathbb{T}_{\tau_i} - \mathbb{P}_{\tau_i}.$$
3) As the other input for $\mathbb{T}_{\tau_{i+1}}$, the arrival count $\mathbb{A}_{\tau_{i+1}}$ of $\tau_{i+1}$ can be inferred by a stochastic process, supposing passengers arrive according to a generic Poisson process.
4) Thus, combining the two equations, we have our key inferring equation in *Dmodel* as follows.
$$\mathbb{T}_{\tau_{i+1}} = (\mathbb{T}_{\tau_i} - \mathbb{P}_{\tau_i}) + \mathbb{A}_{\tau_{i+1}}. \quad (1)$$

*2) Inference Overview:* As in Figure 9, at the end of every time slot, *e.g.*, current time $t_{i+1}$, *Dmodel* infers $\mathbb{T}_{\tau_{i+1}}$ for a segment $s_j$ by Eq.(1) using four steps as follows.

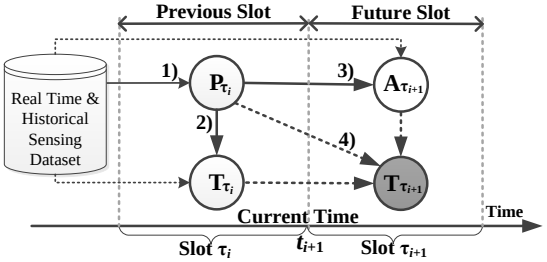


Fig 9: *Dmodel* Inference Overview

1) It *infers* pickup count $\mathbb{P}_{\tau_i}$ by aggregating pickup events in the latest slot $\tau_i$ from *real-time* data;
2) It *infers* total passenger count $\mathbb{T}_{\tau_i}$ based on the corresponding pickup count $\mathbb{P}_{\tau_i}$ and a customized corrective model trained by both *historical* and *real-time* data.
3) It *infers* arrival count $\mathbb{A}_{\tau_{i+1}}$ for the next time slot $\tau_{i+1}$ by a probability distribution $\mathcal{D}$ of passenger arrival rate $\lambda$ at segment $s_j$, which is periodically maintained through a Bayesian updating based on pickup count $\mathbb{P}_{\tau_i}$.
4) It *infers* total passenger count $\mathbb{T}_{\tau_{i+1}}$ for the next slot $\tau_{i+1}$ with Eq.(1), based on inferred $\mathbb{P}_{\tau_i}$, $\mathbb{T}_{\tau_i}$ and $\mathbb{A}_{\tau_{i+1}}$.

In the above steps, steps 1) and 4) are straightforward, so we elaborate steps 2) and 3) in Subsections B and C, respectively.

### B. Inferring Total Passenger Count $\mathbb{T}_{\tau_i}$

In this subsection, we first introduce our key novelty regarding pickup patterns, and then propose how to infer $\mathbb{T}_{\tau_i}$.

*1) Pickup Pattern:* In this work, we infer the total passenger count by four factors, which include (i) *time* in terms of a time slot of a day (*e.g.*, slot $\tau_i$), (ii) *location* in terms of a road segment (*e.g.*, segment $s_j$), (iii) *pickup count* in terms of how many passengers have been picked on a segment during a slot, and (iv) *pickup pattern* in terms of how fast the passengers were picked up, which may include hidden context, *e.g.*, extreme weather or major events. Existing work in the field has considered the first three factors, but the pickup pattern has not been considered by others. In this work, we argue that pickup count is inherently limited by taxicab supply, and cannot provide enough context information to support good inference. However, our pickup patterns provide extra information about hidden context which increases inference accuracy.

Figure 10 presents the same slot $\tau_i$ for two different days with the same pickup count yet with different pickup patterns.

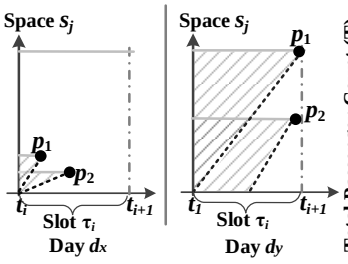


Fig 10: Pickup Patterns

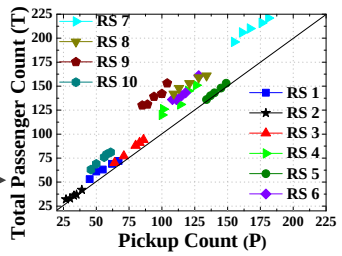


Fig 11: $\mathbb{T}$ vs. $\mathbb{P}$ in 10 Segments

In Figure 10, the key difference of the same slot $\tau_i$ for day $d_x$ and $d_y$ is how long it takes for vacant taxicabs to pick up passengers during $\tau_i$, which is associated to the *pickup pattern*, *i.e.*, the taxicabs on $d_x$ pick up two passengers very quickly; whereas the taxicabs on $d_y$ cruise for a long time before picking up two passengers. The pickup pattern provides extra hidden online context, and cannot be replaced by other contexts already used by existing work, *i.e.*, slot $\tau_i$, segment $s_j$, and pickup count $\mathbb{P}_{\tau_i \in d_z}$ of a particular day $d_z$, since in Figure 10, all other contexts are the same, but two slots $\tau_i$ on $d_x$ and $d_y$ have different pickup patterns. For example, the hidden online context in the pickup pattern during $\tau_i \in d_x$ may indicate suddenly increased passenger demand due to extreme weathers, train arrival or other events, since all taxicabs pick up passengers very quickly. Whereas the pickup pattern for $\tau_i \in d_y$ may indicate a normal scenario without increased demand. Intuitively, though $d_x$ and $d_y$ have same pickup count $\mathbb{P}_{\tau_i \in d_x} = \mathbb{P}_{\tau_i \in d_y} = 2$ (may result from limited taxicab supply), $\tau_i \in d_x$ shall have a larger total passenger count $\mathbb{T}_{\tau_i \in d_x}$ than $\tau_i \in d_y$.

To quantify the pickup pattern as a formal parameter, as in Figure 10, we use the area ratio between the dashed temporal and spatial area (*i.e.*, the union of two dashed triangles) and the total temporal and spatial area (*i.e.*, rectangle) as a *measuring ratio* $\rho$ to show different pickup patterns. The physical meaning of $\rho$ is the ratio between known temporal and spatial inferring area detected by taxicabs and the entire inferring area. As shown in evaluations, $\rho$ accounts for many real-world scenarios that cannot be captured with pickup counts due to the limited taxicab supply.

*2) Customized Online Training:* In what follows, we discuss how to infer the total passenger count based on the new online pickup pattern factor and the other three factors.

Given a segment $s_j$ and a slot $\tau_i$, the pickup count $\mathbb{P}_{\tau_i}$ and the total passenger count $\mathbb{T}_{\tau_i}$ have a logical relationship: $\mathbb{P}_{\tau_i}$ is the lower bound of $\mathbb{T}_{\tau_i}$, since all picked up passengers are included in the total passenger count. Thus, we quantitatively investigate their relationship. Given the historical dataset, for particular slots and segments, we obtain the ground truth of $\mathbb{P}_{\tau_i}$ by aggregated pickup events, and infer the ground truth of $\mathbb{T}_{\tau_i}$ based on the inference of arriving moments (introduced in Section III-B2), *e.g.*, in Figure 7, after inferring arriving moments (shown by stars), $\mathbb{T}_{\tau_i}$ is obtained by counting dashed lines linking dots and stars in a slot (*e.g.*, $\mathbb{T}_{\tau_1} = 3$). Figure 11 gives the relationship between $\mathbb{P}$ and $\mathbb{T}$ for 10 randomly selected road segments in five $\tau_8$ slots from 8AM to 9AM during five weekdays. It indicates an approximate linear relationship between $\mathbb{P}_{\tau_8}$ and $\mathbb{T}_{\tau_8}$ for the same segment $s_j$.

Based on the above observation, we propose a customized online training model based on the linear regression. Suppose that (i) we have a historical dataset consisting of the taxicab GPS data for $K-1$ different days, *i.e.*, day $d_1$ to day $d_{K-1}$, and (ii) the current time is the end of slot $\tau_i$ on day $d_K$, the $Dmodel$ infers the total passenger count $\mathbb{T}_{\tau_i \in d_K}$ as follows.

1) It *calculates* both pickup count $\mathbb{P}_{\tau_i \in d_K}$ and the corresponding measuring ratio $\rho_{\tau_i \in d_K}$, based on the real-time data about the latest slot $\tau_i \in d_K$.
2) It *selects* the data of days whose $\tau_i$ have similar measuring ratio $\bar{\rho}$ to $\rho_{\tau_i \in d_K}$ as a customized training dataset with $M$ pairs of $(\mathbb{P}_{\tau_i \in d_m}, \mathbb{T}_{\tau_i \in d_m})$ where $1 \le m \le M$ (one pair for every day).
3) It *trains* the following model by the $M$ pairs of $(\mathbb{P}_{\tau_i \in d_m}, \mathbb{T}_{\tau_i \in d_m})$ to learn $\alpha_{\tau_i \in d_K}$ and $\beta_{\tau_i \in d_K}$.

$$\mathbb{T}_{\tau_i \in d_m} = \alpha_{\tau_i \in d_K} + \beta_{\tau_i \in d_K} \times \mathbb{P}_{\tau_i \in d_m}. \quad (2)$$

4) It *utilizes* $\alpha_{\tau_i \in d_K}$, $\beta_{\tau_i \in d_K}$ and pickup count $\mathbb{P}_{\tau_i \in d_K}$ to obtain total passenger count $\mathbb{T}_{\tau_i \in d_K}$ with Eq.(2).

A similar measuring ratio $\bar{\rho}$ to $\rho_{\tau_i \in d_K}$ is defined by $\bar{\rho} \in [\rho_{\tau_i \in d_K} - \Delta\rho, \rho_{\tau_i \in d_K} + \Delta\rho]$ where $\Delta\rho$ is a data driven parameter and carefully evaluated in Section V.

### C. Inferring Arrival Count $\mathbb{A}_{\tau_{i+1}}$

The $Dmodel$ infers passenger arrivals with a stochastic process where an arrival rate $\lambda$ of a Poisson Process varies in Brownian motion, which is widely used to model passenger arrival or network packet arrivals [8]. Thus, $\mathbb{A}_{\tau_{i+1}} = \lambda_{\tau_{i+1}} \times |\tau_{i+1}|$. Note that we did not use customized training to infer the arrival count $\mathbb{A}_{\tau_{i+1}}$ based on given pickup count $\mathbb{P}_{\tau_i}$ as in the last subsection, since there is no potentially logical relationship between $\mathbb{P}_{\tau_i}$ and $\mathbb{A}_{\tau_{i+1}}$.

*1) Passenger Arrival Rate Modeling:* The $Dmodel$ maintains a probability distribution $\mathcal{D}$ of $\lambda$ for a segment $s_j$ by discretizing the space of passible $\lambda$, and assumes that (i) $\lambda$ is one of a set of discrete values from 0 to the maximum $\lambda$ (obtained by the dataset) and (ii) the initial probability for all possible $\lambda$ are uniformly distributed. Therefore, at the end of the slot $\tau_i$, $Dmodel$ updates $\mathcal{D}$ with three steps.

1) It *evolves* $\mathcal{D}$ to the current time by applying Brownian motion to every possible rate.
2) It *infers* the arrival count $\mathbb{A}_{\tau_i}$ in $\tau_i$ based on observed $\mathbb{P}_{\tau_i}$, and *calculates* probabilities that this arrival count $\mathbb{A}_{\tau_i}$ is associated to every one of arrival rates as follows.

$$F(x) \leftarrow \mathcal{D}_{\text{old}}(\lambda_{\tau_i} = x) \times \mathrm{e}^{-x \cdot |\tau_i|} \frac{(x \cdot |\tau_i|)^{\mathbb{A}_{\tau_i}}}{\mathbb{A}_{\tau_i}!}.$$

3) It *normalizes* these probabilities, so they sum to unity.

$$\mathcal{D}_{\text{new}}(\lambda_{\tau_i} = x) \leftarrow \frac{F(x)}{\sum_k F(k)}.$$

These three steps constitute Bayesian updating for $\mathcal{D}$. Given $\mathcal{D}$, we infer $\lambda_{\tau_{i+1}}$ with a cautious estimate to bound a risk of overinferring. So, we employ the $\omega$th percentile of $\mathcal{D}$ to calculate the inferred $\lambda_{\tau_{i+1}}$, *e.g.*, 40th percentile. In $Dmodel$, $\omega$ is a data driven parameter, and is evaluated in Section V.

A key unresolved question is how to infer the arrival count $\mathbb{A}_{\tau_i}$ by the pickup count $\mathbb{P}_{\tau_i}$, which is introduced as follows.

*2) Inferring Previous Arrival Count $\mathbb{A}_{\tau_i}$:* We introduce how to infer $\mathbb{A}_{\tau_i}$ in Figure 12 where we classify all passengers associated to the total passenger count $\mathbb{T}_{\tau_i}$ of a slot $\tau_i$ into four parts, based on when they arrived and whether they get picked up at the end of slot $\tau_i$. Thus, the sum of passengers in Part 1 and Part 2 is the arrival count $\mathbb{A}_{\tau_i}$ we try to infer. The sum of passengers in Part 2 and Part 3 is the pick up count $\mathbb{P}_{\tau_i}$; the sum of passengers in all four parts is the total passenger count $\mathbb{T}_{\tau_i}$; we have already obtained both of them in the previous subsection.

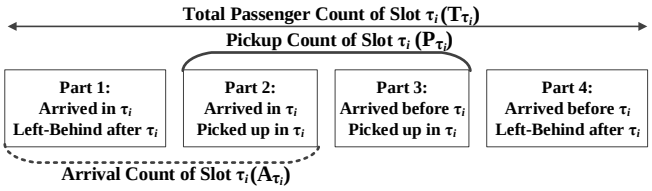


Fig 12: Inferring Previous Arrival Count $\mathbb{A}_{\tau_i}$

We add the following two kinds of passengers to infer $\mathbb{A}_{\tau_i}$.

1) *Passengers in Part 1*: Since we have already inferred the total passenger count $\mathbb{T}_{\tau_i}$ based on $\mathbb{P}_{\tau_i}$ in Section IV-B, we have the total number of passengers in Part 1 and 4 together, *i.e.*, $\mathbb{T}_{\tau_i} - \mathbb{P}_{\tau_i}$. Further, since an inferring slot (*e.g.*, one hour) is typically longer than a passenger waiting period, the number of passengers in Part 4 is 0. Thus, we have the number of passengers in Part 1 alone.
2) *Passengers in Part 2*: We can differentiate the passengers in Parts 2 and 3 by inferring arriving moments of these picked up passengers in $\mathbb{P}_{\tau_i}$, based the method in Section III-B2, and then we obtain the number of passengers in Part 2 alone.

## V. $Dmodel$ Evaluation

We evaluate $Dmodel$ based on the 450 GB dataset introduced in Section III. In this large dataset, we mainly find two kinds of errors. (i) Missing Records: a fair amount of sensing records are missing. (ii) Location Errors: GPS coordinates indicate that a taxicab is off road. Different reasons, *e.g.*, device malfunctions, can lead to such errors. We perform a preprocessing to clean this dataset to rule out taxicabs with more than $10\%$ of errant records.

### A. Evaluation Methodology

We divide the entire 182 days dataset into two subsets. *Testing Dataset*: it contains the data about a particular day, *e.g.*, day $d_1$, serving as the real-time streaming data in the evaluation. *Training Dataset*: it contains the data about the rest of days, serving as the historical training data. For this particular day $d_1$, if we use one hour slots, at the end of the first slot, *i.e.*, time 01:00, we use $Dmodel$ to infer the total passenger count for the next slot from 01:00 to 02:00, based on both the "real-time" data from 00:00 to 01:00 in the testing dataset, and all data in the historical training data. We let the testing dataset rotate among all 182 days of data, leading to 182 sets of experiments. The average results are reported.

We compare $Dmodel$ with two models: SDD and Basic. **SDD** is one of the state-of-the-art taxicab demand and supply models, which maintains a distribution for passenger demand based on the previous average demand [4]. SDD serves as a statistical model and is suitable for the real world scenario where the real-time data collection is not possible, and we can only use the historical data to infer passenger demand. **Basic** model first uses a generic offline training to train the entire dataset to obtain parameters ($\alpha$ and $\beta$) in offline without considering the pickup pattern. Basic serves as a baseline for $Dmodel$ to show the effects of the ignorance of logical contexts shown by the pickup patterns (*e.g.*, extreme weathers or events) on the model performance. $Dmodel$ performs similarly with Basic except that it uses logical contexts (pickup and cruising events) in the testing dataset to calculate a measuring ratio for a particular slot, and selects the data of slots with similar pickup patterns (shown by measuring ratios) as a customized training dataset to perform an online training as introduced in Section IV-B2.

By processing the entire dataset with the method of inferring the passenger arrival moments (introduced in Section III-B2), we infer the ground truth of the total passenger count used to test models with a key metric, called **Inference Accuracy**. The inference accuracy is defined as a ratio equal to $= \frac{\bar{\mathbb{T}} - |\bar{\mathbb{T}} - \mathbb{T}|}{\bar{\mathbb{T}}}$ where $\mathbb{T}$ is the inferred total passenger count of a particular model and $\bar{\mathbb{T}}$ is the total passenger count obtained from the inferred ground truth.

We first test models on 1000 road segments about accuracy with different slot lengths. Then, we investigate the sensitivity of $Dmodel$ to two key parameters: $\Delta\rho$ and $\omega$ used in Sections IV-B2 and IV-C1, and obtain their optimal default values.

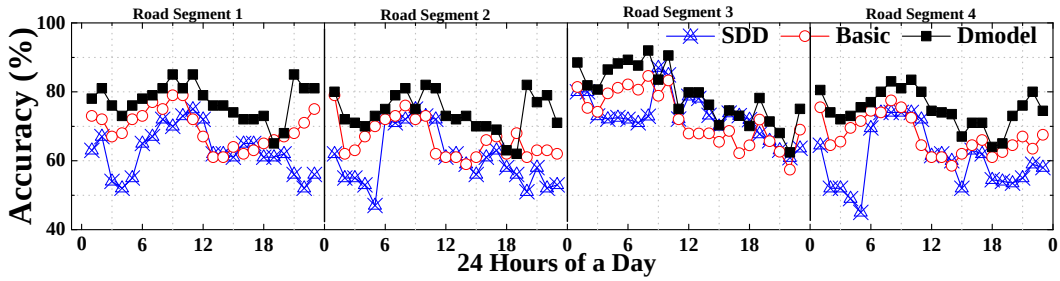


Fig 13: Accuracy under One Hour Slot for 24 Hours of a Day in Four Road Segments

### B. Inference Accuracy

In this subsection, we compare three models' inferring accuracy in different lengths of slots. We show low level comparisons on 4 particular road segments, and high level comparisons on 1000 road segments. All road segments are randomly selected in the downtown area of Shenzhen.

*1) Low Level Comparisons:* Figure 13 plots the accuracy of three models on 4 road segments under one hour slots. $Dmodel$ has better performance than Basic and SDD, especially at the non-rush hour, *e.g.*, 18:00 to 06:00. Basic outperforms SDD in the early morning, *e.g.*, 00:00 to 06:00, and the late night, *e.g.*, 18:00 to 00:00. SDD has a good accuracy during the morning rush hour, *e.g.*, 08:00 to 12:00, and we believe this is because during the rush hour, passenger demand is relative stable compared to other times.

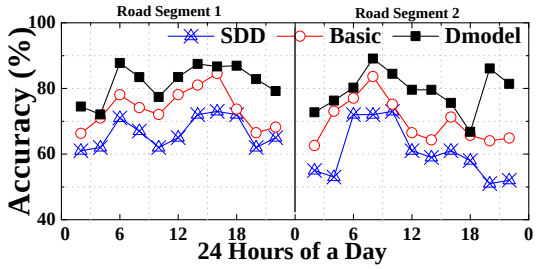


Fig 14: Accuracy under Two Hour Slots for 24 Hours of a Day

Figure 14 shows comparisons in segment 1 and 2 under two hour slots. With a longer slot, the accuracy generally increases for all three models. This is because (i) passenger demand is more stable in a longer slot, and thus SDD model becomes more effective; (ii) a longer slot increases accuracy of passenger arrival predictions in $Dmodel$ and Basic, which leads to increased inference accuracy.

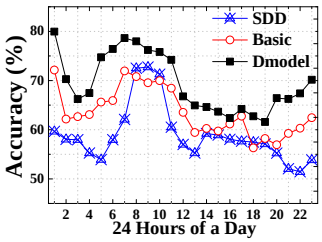


Fig 15: Hourly Average Accuracy

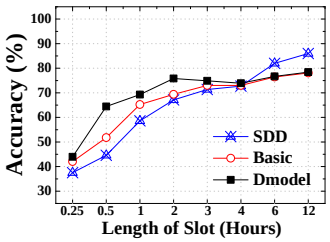


Fig 16: Effects of Slot Lengths

*2) High Level Comparisons:* Figure 15 gives the average accuracy on 1000 road segments under one hour slots at different hours of a day. The average accuracy of all three models on 1000 road segments is lower than the accuracy we observed in 4 particular road segments. It is because the passenger demand may change dramatically between different segments. But the relative performance between three models is similar to Figure 13. $Dmodel$ has a better performance than SDD and Basic by 39% and 13% on average, which results from its customized online training.

Figure 16 gives the average accuracy on 1000 segments with different slot lengths. The average accuracy of all models increases with the lengths of slots. SDD outperforms Basic and $Dmodel$ when slots are longer than 6 hours. This is because the passenger demand in a longer slot becomes more stable on different days. When the slot becomes longer, $Dmodel$ and Basic have the similar performance, because measuring ratios for long slots are mostly equal, and cannot be used by $Dmodel$ to differentiate related time slots.

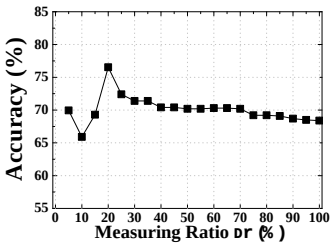


Fig 17: $\Delta\rho$ vs. Accuracy

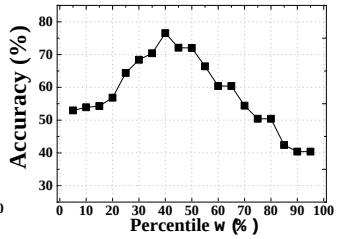


Fig 18: $\omega$ vs. Accuracy

### C. Sensitivity of Dmodel

We investigate the sensitivity of $Dmodel$ to two parameters $\Delta\rho$ and $\omega$ on 1000 road segments under two hour slots.

*1) $\Delta\rho$ vs. Accuracy:* $\Delta\rho$ is used to decide the similarity between measuring ratios as in Section IV-B2. Figure 17 gives effects of $\Delta\rho$ on $Dmodel$. With the increase of $\Delta\rho$, the accuracy of $Dmodel$ increases first, and then decreases. This is because when $\Delta\rho$ increases, $Dmodel$ finds more slots with similar pickup patterns to effectively train customized corrective model online. But when $\Delta\rho$ becomes too large, $Dmodel$ has to consider more slots with different pickup patterns, leading to poor performance. The accuracy peaks when $\Delta\rho = 0.2$, which is set as the default value of $\Delta\rho$. If the used $\Delta\rho$ leads to an empty training dataset for $Dmodel$, $\Delta\rho$ increases until the training dataset is no empty.

*2) $\omega$ vs. Accuracy:* $\omega$ is used to decide the percentile to predict the future passenger arrival as in Section IV-C1. Figure 18 plots effects of $\omega$ on $Dmodel$. A small $\omega$ indicates that $Dmodel$ conservatively predicts the arrival rate; whereas a large $\omega$ indicates that $Dmodel$ aggressively predicts the arrival rate. We find that either a small or a large $\omega$ leads to poor performance. The accuracy peaks when $\omega = 0.4$, which is set as the default value of $\omega$.

## VI. *Dmodel* Application

We propose a *Dmodel* application where a dispatching center employs demand inferred by *Dmodel* to achieve an equilibrium between passenger demand and taxicab supply, based on a complete data driven design given 245 urban regions Shenzhen as shown in Figure 4.

In our application, at the end of a "real-time" slot $\tau_i$, we first use *Dmodel* to infer passenger demand $\mathbb{T}^{r_x}_{\tau_{i+1}}$ for the next slot $\tau_{i+1}$ in a region $r_x$ by aggregating inferred demand of all segments in $r_x$. Next, we employ real-time data to aggregate vacant taxicabs in region $r_x$ to obtain "dispatchable" vacant taxicab supply in $r_x$ for the next slot $\tau_{i+1}$, indicated by $\mathbb{S}^{r_x}_{\tau_{i+1}}$. Similarly, we shall have all $\mathbb{T}^{r_x}_{\tau_{i+1}}$ and $\mathbb{S}^{r_x}_{\tau_{i+1}}$ where $1 \leq x \leq 245$. Finally, we use a straightforward scheme to dispatch taxicab supply $\sum_{1\leq x\leq 245} \mathbb{S}^{r_x}_{\tau_{i+1}}$ among 245 regions so that the dispatched taxicab supply $\hat{\mathbb{S}}^{r_x}_{\tau_{i+1}}$ for a region $r_x$ is proportional to inferred demand $\mathbb{T}^{r_x}_{\tau_{i+1}}$ in the region $r_x$.

The evaluation is based on the ground truth of passenger demand $\bar{\mathbb{T}}^{r_x}_{\tau_{i+1}}$ of slot $\tau_{i+1}$, and the dispatched vacant taxicab supply $\hat{\mathbb{S}}^{r_x}_{\tau_{i+1}}$ in each region at hourly slots. We propose a normalized equilibrium value $0 \leq \kappa \leq 1$ to evaluate effectiveness of dispatching: $\kappa_{\tau_{i+1}} = \arg_{1\leq x\leq 245} \frac{|\bar{\mathbb{T}}^{r_x}_{\tau_{i+1}} - \hat{\mathbb{S}}^{r_x}_{\tau_{i+1}}|}{\bar{\mathbb{T}}^{r_x}_{\tau_{i+1}} + \hat{\mathbb{S}}^{r_x}_{\tau_{i+1}}}$. If the demand inferred by an inference method, *e.g.*, *Dmodel*, is similar to the ground truth of the demand, the corresponding dispatch leads to a small $\kappa$, indicating an equilibrium between passenger demand and taxicab supply; otherwise, it leads to a large $\kappa_{\tau_{i+1}}$, *i.e.*, disequilibrium. Note that dispatching taxicabs skews the historical dataset in terms of taxicabs' traces. To eliminate dispatching effects, we only used dispatched supply to calculate $\kappa$, and did not manipulate taxicabs' traces, and then we started over at the end of the next slot.

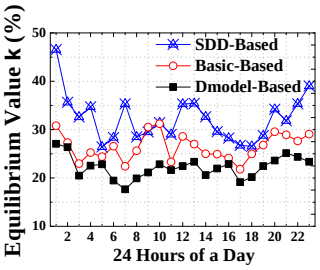


Fig 19: $\kappa$ under one hour slots

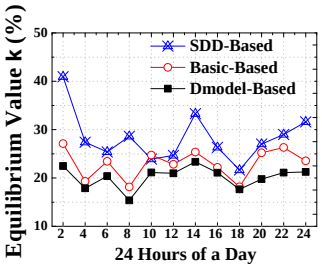


Fig 20: $\kappa$ under two hour slots

Figure 19 plots the equilibrium value $\kappa$ at different hours of a day under one hour slots. We observe that the equilibrium values fluctuate in all dispatching. But *Dmodel* based dispatching has a lower equilibrium value almost at every slot. Basic based dispatching outperforms SDD based dispatching at the most of the time. Figure 20 gives the average equilibrium value $\kappa$ under two hour slots. We find the equilibrium values under two hour slots are lower than the equilibrium values under one hour slots for all dispatching, which verifies our previous observation that two hour slot based inferring is better than one hour slot based one in terms of the inference accuracy.

## VII. Related Work

Several novel systems have been proposed using taxicab traces obtained from on-board GPS devices. For example, some systems are proposed to assist taxicab operators better oversee taxicabs and to provide timely services to passengers, *e.g.*, predicting passenger demand [3] [4] [5], detecting anomalous taxicab trips to discover driver fraud [9], and discovering temporal and spatial causal interactions to provide timely and efficient services in certain areas with disequilibrium [10]. In addition to taxicab operators, several systems are proposed for the benefit of passengers or drivers, *e.g.*, allowing taxicab passengers to query the expected duration and fare of a planed trip based on previous trips [11] and estimating city traffic volumes for drivers [12].

However, our model is different from the existing research by its novel inference method based on real-time and historical data from roving sensor networks. Technically, we focus on inferring passenger demand based on a customized online training method utilizing real-time pickup patterns and hidden contexts (*e.g.*, arriving moments) obtained by roving taxicab sensors, which has not been investigated before.

## VIII. Conclusion

Based on a 450 GB dataset, we design and evaluate a taxicab passenger model *Dmodel* with a 76% inference accuracy. Our effort provides a few valuable insights. Specifically, (i) the mobile taxicabs can be used as roving sensors to infer passenger demand with high accuracy; (ii) the inference accuracy is highly depended on locations, times, and other logical information; (iii) the length of inferring slots has significant impacts on the inference accuracy, and a good tradeoff between usability and accuracy of inference methods has to be carefully evaluated; (iv) a statistical passenger demand model can be enhanced by generic offline training that takes the waiting passengers into accounts, but it can be further enhanced by a customized online training for particular real-time situations, shown by our 39% performance gain.

## References


[1] "National transport authority," in *National Taxi Fare Review 2012*.

[2] "San francisco municipal transportation agency:taxi user surveys," in *San Francisco Municipal Transportation Agency*.

[3] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, "An energy-efficient mobile recommender system," in *KDD '10*.

[4] Y. Huang and J. W. Powell, "Detecting regions of disequilibrium in taxi services under uncertainty," in *SIGSPATIAL '12*.

[5] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger," in *UbiComp '11*.

[6] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *KDD '11*.

[7] S. Standard, "Shenzhen ranks fifth in the world in terms of population density," *http://www.shenzhen-standard.com*.

[8] V. Paxson and S. Floyd, "Wide area traffic: the failure of poisson modeling," *IEEE/ACM Trans. Netw.*

[9] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "ibat: detecting anomalous taxi trajectories from gps traces," in *UbiComp '11*.

[10] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," in *KDD '11*.

[11] R. K. Balan, K. X. Nguyen, and L. Jiang, "Real-time trip information service for a large taxi fleet," in *MobiSys '11*.

[12] J. Aslam, S. Lim, X. Pan, and D. Rus, "City-scale traffic estimation from a roving sensor network," in *SenSys '12*.