Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 5414930, 18 pages
https://doi.org/10.1155/2017/5414930

Research Article

A Real-Time Taxicab Recommendation System Using Big Trajectories Data

Pengpeng Chen, Hongjin Lv, Shouwan Gao, Qiang Niu, and Shixiong Xia

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China

Correspondence should be addressed to Shixiong Xia; xiasx@cumt.edu.cn

Received 8 January 2017; Revised 24 May 2017; Accepted 12 June 2017; Published 25 July 2017

Academic Editor: Paolo Bellavista

Carpooling is becoming a more and more significant traffic choice, because it can provide additional service options, ease traffic congestion, and reduce total vehicle exhaust emissions. Although some recommendation systems have proposed taxicab carpooling services recently, they cannot fully utilize and understand the known information and essence of carpooling. This study proposes a novel recommendation algorithm, which provides either a vacant or an occupied taxicab in response to a passenger's request, called VOT. VOT recommends the closest vacant taxicab to passengers. Otherwise, VOT infers destinations of occupied taxicabs by similarity comparison and clustering algorithms and then recommends the occupied taxicab heading to a close destination to passengers. Using an efficient large data-processing framework, Spark, we greatly improve the efficiency of large data processing. This study evaluates VOT with a real-world dataset that contains 14747 taxicabs' GPS data. Results show that the ratio of range (between forecasted and actual destinations) of less than 900 M can reach 90.29%. The total mileage to deliver all passengers is significantly reduced (47.84% on average). Specifically, the reduced total mileage of nonrush hours outperforms other systems by 35%. VOT and others have similar performances in actual detour ratio, even better in rush hours.

1. Introduction

Urban air and soil quality are essential to the health of urban residents. Good urban air and soil quality can greatly improve the function of the nervous system, enhance the efficiency of work, and ensure the healthy status of urban residents [1]. However, taxicab exhaust emissions have an extremely negative effect on urban soil [2] and air quality [3]. In Beijing, a taxi can run hundreds of thousands of kilometers a year [4, 5]. Under normal circumstances, exhaust emission from a taxi is more than 5 times the emission from a private car.

Carpooling services can effectively reduce the excessive emissions from taxis by reducing the total mileage to deliver all passengers. But unlike regular taxicab services that arbitrarily assign one vacant taxicab to a new passenger [6, 7], taxicab carpooling services require catching a particular taxicab, which refers to a taxicab with existing passengers heading to a direction similar to that of the new passenger. However, the occupied taxicab could not be found for a passenger based on the existing solutions for finding a vacant taxicab.

For the carpool service, there are mainly two categories: static and dynamic carpooling. In the static carpooling research, most researches focus on how passengers with similar destinations are assigned to a car [8–10] and how to improve the timeliness for the real-time performance of the carpooling service [11–13]. In all, the static carpooling problem in a sense can be regarded as a special member of the general class of the Dial-a-Ride Problem (DARP) [14].

Although the static carpooling researches have greatly improved the performance of carpooling services, the above researches are all built on the premise that the information of all passengers is known in advance. But the travel routes and time of existing passengers in taxicabs are not accessible for us on the basis of the existing infrastructure, unless we spend a huge fortune building a new thorough taxi system. In addition, the size of increasing vehicle data goes far beyond the range of DARP. Since the general DARP is NP-hard [15], only small datasets can be dealt with optimally [16, 17]. However, the further development of big-data-processing technology and the upgrading of taxi equipment (GPS [18] and fare meters), forming a huge GPS records database

with rich semantic information, provide an opportunity for predicting existing passengers' information, namely, the core of dynamic carpooling.

This paper belongs to the dynamic carpooling research. In dynamic carpooling, we do not have any information about travel routes and travel time of passengers in advance. What is more, reasonable request matching needs to be timely and efficiently accomplished with continuous query requests generated in real time. Thus, dynamic carpooling has the characteristics of real time, quick response, reasonable matching, and so forth. These characteristics are undoubtedly quite suitable for the large-scale taxi scene and more in accord with the needs of the public. Thus, this paper focuses on real-time dynamic carpooling based on taxicabs' GPS records.

Based on big-data-processing technology and historical taxicab GPS data, some researches [19, 20] provide a dynamic real-time carpooling service. However, existing dynamic carpooling researches have four defective aspects: (I) inadequate information mining, (II) ignoring valuable situations, (III) ignoring destination distribution characteristics, and (IV) one-sidedness of screening criteria. In Section 2, we will elaborate on these defective aspects and propose our motivations.

In this study, we propose *VOT*, a taxicab recommendation system based on extremely large taxicab GPS data. By using a unified standard to distinguish taxicab performances, *VOT* provides both carpooling and conventional taxicab services, which can effectively reduce the excessive emissions of taxis. The key contributions of this study are as follows:

(i) To the best of our knowledge, we propose the first carpool service, which can significantly reduce the total mileage to deliver all passengers under the premise of fully ensuring the interests of passengers. In addition, for raw GPS datasets with unstructured format, Spark is applied to improve the efficiency of large data processing.

(ii) To achieve our goal, we design a novel method to predict the occupied taxicabs' destination by similarity comparison and clustering algorithms. It can obtain more accurate forecasting destinations by fully mining GPS datasets and eliminating interferences from worthless destinations.

(iii) To more comprehensively evaluate the taxicab carpooling performance, we further propose a novel metric called Distance Dispersion, which is defined as an average distance between a particular passenger's destination and possible destinations of occupied taxicabs.

(iv) We evaluate *VOT* with a real-world dataset, containing 14747 taxicabs' GPS data. The results show that the ratio of range (between forecasted and actual destinations) of less than 900 M can reach 90.29% and *VOT* can reduce 53% of the total mileage to deliver all passengers, especially outperforming other systems by almost 35% at 0:00 to 7:00 AM.

The rest of the paper is organized as follows. Section 2 introduces our motivation. Section 3 presents taxicab networks research. Section 4 proposes our system overview. Section 5 depicts the system implementation. Section 6 validates our design with datasets. Several practical issues are discussed in Section 7, followed by the conclusion in Section 8.

2. Motivation

In this section, we present our motivations to improve the four legacy defects for taxicab carpooling services based on empirical data from a real-world taxicab network of 14747 taxicabs in Shenzhen [21].

First, we demonstrate theoretically four defects in the existing dynamic carpooling system and then further clearly interpret these deficiencies by figures and experiments. Finally, we discuss the methods we adopt to make up for these weaknesses.

2.1. Inadequate Information Mining. In dynamic carpooling services, we need to predict the potential destinations of these real-time occupied taxicabs for detecting this one with the best carpooling performance. However, we argue that although the potential destinations would be obtained eventually, little information (only the origin of occupied taxicabs and real-time passengers) is used to predict destinations in existing dynamic carpooling research. In other words, the potential destinations are inferred by finding similar trajectories that start at the same origin (the real-time occupied taxi) and pass the same location (starting point of passenger P) in other researches.

As shown in Figure 1(a), the passenger P sends a carpool request to the server at O_P. At this point, the real-time occupied taxi T (taking the existing passenger on O_T, passing through L1, L2, L3, ..., L9 to an unknown destination) in L9 can serve as a potential carpooling option for the passenger P, so we need to infer the destination of T (or the existing passenger) for quantifying its carpool performance.

As shown in Figure 1(b), existing dynamic carpooling studies use only C1 (the nearest intersection from T's origin O_T) and C4 (the nearest intersection from P's origin O_P) as the matching criteria. This approach ignores the valuable information between O_T and O_P. To the best of our knowledge, the more detailed the matching data we provide is, the more accurate our matching results will be. Therefore, the method of applying the last manned trajectory (between O_T and O_P) of T as the matching data in *VOT* is a necessary supplement to higher forecasting precision.

2.2. Ignoring Valuable Situations. The application of only two origins (real-time occupied taxicabs and passengers) not only results in the incomplete mining of the GPS dataset, but also ignores a great deal of valuable historical trajectory information (with high similarity).

Compared with the last manned trajectory of real-time taxicabs, the historical manned trajectories, especially with a higher degree of similarity, have a higher likelihood of having the same destination as these real-time taxicabs. Because

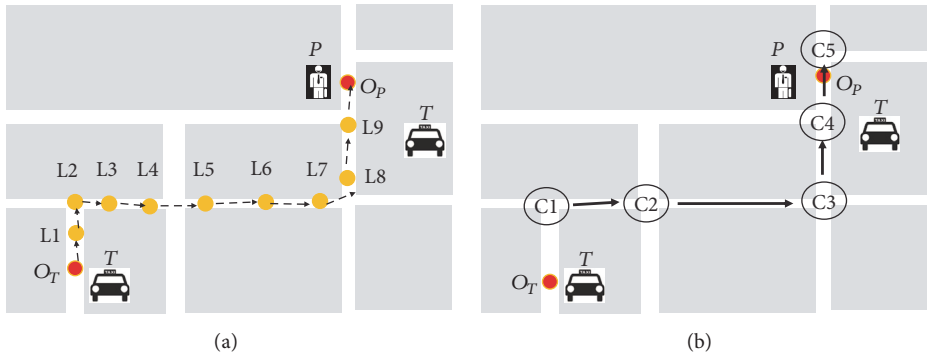


Figure 1: Real GPS records (VS) existing methods for dynamic carpooling.

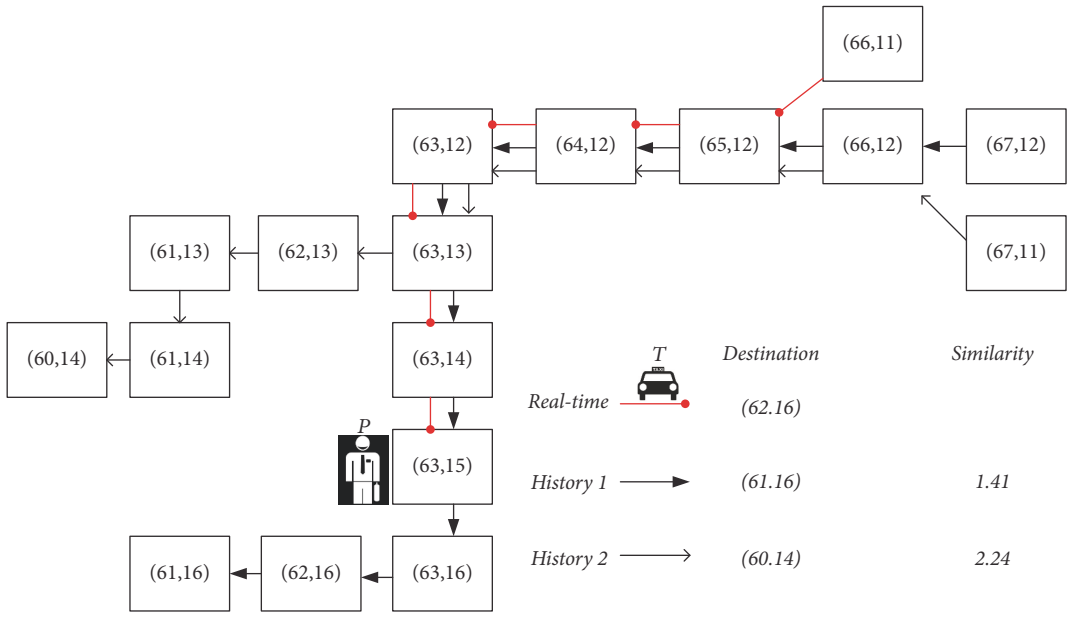


Figure 2: Ignoring valuable situations.

two pairs of latitude and longitude with the same value hardly exist, we introduce the regional division in this paper. Then, the map is divided into many marked regions (two-dimensional value, like (63, 15)).

We extract a case from experiments applying real GPS records and draw it up in Figure 2. As shown in Figure 2, when a passenger P asks for a carpooling request in (63.15) region, there is a real-time occupied taxicab T that can be regarded as a potential carpool option. Then, *VOT* puts the last manned trajectory data of T as matching data and compares it with the historical manned trajectory dataset. Compared with "History 2," the destination of "History 1" with greater similarity is closer to the destination of T. This confirms our previous conclusions.

2.3. Ignoring Destination Distribution Characteristics. After obtaining the initial potential destinations collection, existing dynamic carpooling schemes equally treat all destinations that appear in this collection and regard the frequency of potential destinations as their probability. However, there are two drawbacks if we follow the existing methods:

(A) Existing researches ignore the fact that the historical trajectories that generate the preliminary potential destinations collection have different similarity. In other words, each possible destination corresponds to different possibilities (by quantifying the similarity of historical trajectories). Compared with existing studies, this paper aims to detect those destinations with high similarity and high frequency, instead of only focusing on frequency.

(B) In the existing dynamic carpooling, a large proportion is allocated to massive possible destinations with quite low frequency. It has little chance to be the real destination when the region has few frequencies. Moreover, existing studies ignore the characteristics of destination distribution with regional distribution [22–24]. In other words, the vast majority of destinations are distributed in several hot spots.

Considering the above-mentioned limitations, we concentrate our efforts on finding the most likely regions and try our best to eliminate the interference of loose and extremely

low frequency destinations. Therefore, *VOT* makes use of clustering algorithms to divide the potential destinations and applies the cluster center to represent all the possible destinations in the same cluster. In this way, it highlights these regions with high frequency and high similarity to the greatest extent. What is more, even if the true destinations of real-time occupied taxicabs are not the forecasted cluster centers, the distance between these true destinations and cluster centers is quite small. To validate our design, we propose a new parameter in Section 6, called Real Prophecy Distance (RPD), to test *VOT* on the entire GPS dataset.

2.4. One-Sidedness of Screening Criteria. In this work, we argue that although carpooling choice would be obtained eventually, the ultimate carpooling choice should not be obtained by a parameter that can only meet the requirements of carpooling service in one side. Taxicab GPS records have been used by several systems to provide dynamic carpooling services. But existing researches, which mainly focus on detour distance, cannot perform well in both the interests of passengers and the mitigation of gas exhaust emissions.

As well known, if carpooling passengers have the same destination as the existing passengers, they would debus at the same time and place. Under this scenario, the carpooling service achieves its best utility, in which the carpooling passengers have no detour distance. Meanwhile, it reduces the mileage of the whole trip of carpooling passengers. In other words, a greater degree of closeness between the carpooling passengers' destination and the destinations of occupied taxicabs indicates lower extra consumption and a better carpooling performance.

Thus, we conduct our first work to provide carpooling services, which applies a novel parameter called Distance Dispersion to quantify the closeness between the destinations of particular passengers P and occupied taxicabs. The ultimate carpooling strategy for P in this paper is to select an occupied taxicab with the minimum Distance Dispersion as the "can-carpool" taxicab. In order to prove the superiority of Distance Dispersion, we evaluate the performance of *VOT* through actual detour ratio (%) and reduced total mileage (%) in Section 6.

3. Taxicab Networks Infrastructure

In this section, we present the taxicab networks infrastructure and the implicit semantic information inferred from the raw large GPS dataset.

3.1. Infrastructure. Underlying taxicab infrastructures in large cities are presently equipped with GPS, communication devices, and dispatch centers. Based on the upgrades of taxicab devices, the taxicab network can be roughly divided into two parts, namely, (1) numerous taxicabs, in the frontend, which provide service and assume the role of the sensing terminal at the same time, and (2) dispatching centers with cloud servers, in the backend, to receive and store sensing records for the taxicab service [25, 26].

The establishment of the large taxi GPS dataset is the foundation of system implementation. Based on the popularity of taxicabs' underlying infrastructure, these locations and statuses are periodically uploaded to the dispatching center, which forms a large taxi GPS dataset. The formation step of this dataset is presented as follows:

(1) Loaded with a wireless transmission module, the taxicab would cyclically send its status to the nearest cell tower.

(2) The status data would be forwarded to the cloud server by the cell tower.

(3) The real-time GPS data are stored in the cloud server established for analysis according to the fixed format.

Each GPS record of the large GPS dataset contains all the attribute categories of the taxicab real-time information. A GPS record mainly consists of the following parameters: plate number, which is the unique identification of taxicabs; date and time, which demonstrate the time of this record generated by the GPS device; GPS coordinates, which monitor the global status of the taxicab; Status Bit, which indicates if some passengers exist when this record is uploaded.

Real-time GPS records of tens of thousands of taxicabs would be uninterruptedly transmitted to the cloud server, forming large amounts of GPS trajectory information. Such raw large GPS dataset has a very high resolution, which can be used to locate a particular taxicab at fine granularity related to both time and space. Nonetheless, such a fine-granular large GPS dataset has many erroneous and missing records. Meanwhile, such a raw GPS dataset could not be obtained firsthand as it is in a format that is not ready for analysis [27]. In the next subsection, we extract useful implicit semantic information about the taxicab service from the raw large dataset.

3.2. Implicit Information in Underlying Infrastructure. Based on historical and real-time GPS records, we observe four statuses related to passenger demand by continuously tracking the GPS records of the same taxi.

(1) *Take-In Status.* For the same taxicab, if its status value turns from "0" to "1" in two consecutive records, then this taxi just picked up a passenger. The location of Take-In Status is considered an origin or a take-in location of a trip.

(2) *Drop-Off Status.* If the status value turns from "1" to "0" in two successive records, then this taxicab just dropped off a passenger. The location of Drop-Off Status is considered a destination of a trip.

(3) *Occupied Status.* Continuously observing the same taxi, if the status value keeps "1," then the taxi is heading to the destination of the passengers. We believe the location of Occupied Status is the middle section of one trajectory.

(4) *Wander Status.* When we continuously observe the GPS records of taxicabs, the taxi is at Wander Status if the status value holds "0" all the time.

Based on the implicit semantic information mined from the real-time GPS dataset, the regular taxicab recommendation systems can efficiently locate and recommend vacant

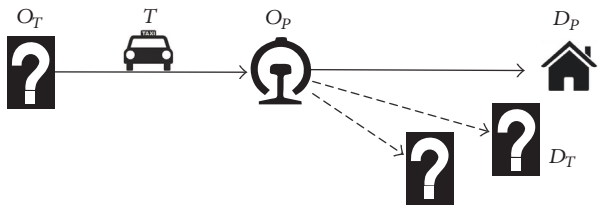


Figure 3: Semantic demonstration.

taxicabs to real-time particular passengers. Some existing recommendation systems even could provide a carpooling option when no nearby vacant taxicab is available. But they fail to guarantee result accuracy because of the low utilization of the large dataset and the inference from numerous worthless destinations with low frequency and low similarity. What is more, existing dynamics carpooling researches ignore the characteristics of destination distribution with regional distribution and cannot perform well in both the interests of passengers and the mitigation of gas exhaust emissions and traffic congestion.

Our extensive understanding on the large GPS dataset and carpooling service provides an opportunity to obtain higher inference accuracy. Based on the above analysis and discussion, our recommendation system locates and recommends the best taxicab in the performance of carpooling and conventional service to the real-time passenger, which is presented in the next section.

4. System Overview

This recommendation system is designed to mine GPS records in depth for enhanced recommendation quality. Considering that regular services are commonly understood, we provide a scenario in which carpooling services are applied, and then we present the main idea of our recommendation system.

4.1. Scenario Demonstration. Figure 3 presents a scenario in which passenger P requests for a taxi at origin (O_P) heading to destination (D_P). Built on the implicit semantic information in underlying taxicab infrastructure and specific passenger information, no taxis in the Wander Status are found around the passenger P. But, based on the observation on real-time GPS records, the recommendation system can locate nearby occupied taxi T as a potential "can-carpool" taxicab (heading to an unknown destination) that will pass the origin of P soon. Owing to the limited knowledge on the destinations of existing passengers on taxicab T, carpool service could not be reached just with the request of passenger P.

By reverse tracking on the real-time GPS records based on time, *VOT* obtains the last manned trajectory (between O_T and O_P) of T. Compared with this last manned trajectory, the historical trips, especially with a higher degree of similarity, have a higher likelihood of having the same destination as the existing passengers. Thus, *VOT* fully mines the historical and real-time GPS records and regards the destinations of highly similar historical trajectories as potential destinations.

VOT further optimizes the potential destination sets by the clustering algorithm catching center regions, which can efficiently summarize the features of destination distribution and thoroughly reduce the interference from worthless destinations with low frequency and similarity. In this study, we catch these center regions by using different clustering algorithms (K-means [28, 29], density-based spatial clustering of applications with noise (DBSCAN) [30, 31], and balanced iterative reducing and clustering using hierarchies (BIRCH) [32, 33]).

When a nearby occupied taxicab provides a carpooling service to the particular passenger P, the real trip of P generates additional consumption compared with the conventional taxi service. Therefore, the optimal carpooling strategy means a "can-carpool" taxi with the lowest consumption. A greater degree of closeness between the destinations of carpooling passengers and occupied taxicabs indicates lower consumption and a better carpooling performance.

Therefore, a novel parameter called Distance Dispersion is used to quantify the degree of closeness in *VOT*. Distance Dispersion could be obtained by averaging the Manhattan and the Euclidean Distances between the real-time passengers' destination and forecasted potential destinations. Different occupied taxicabs have different destinations, resulting in different Distance Dispersions for P to carpool. The optimal carpooling strategy for P is to select an occupied taxicab with the least Distance Dispersion as the "can-carpool" taxicab.

4.2. Main Procedure. The main procedure of *VOT* is presented in Figure 4.

4.2.1. Manned Trajectory Distributions. The taxicab manned trajectory distribution, which is the foundation of carpooling service, plays a crucial role in our recommendation system.

We separate individual trips from the entire historical GPS dataset by continuously tracking and observing the change in Status Bit on the GPS records of the same taxicab. The distribution, generated from the large GPS dataset, contains historical GPS records for all taxicabs. With the context of a particular passenger, such a distribution can generate the potential destinations of trajectories with a high degree of similarity compared to another certain trajectory.

4.2.2. Distance Dispersion Calculation. Based on the manned trajectory distribution, when receiving a request from passenger P, our recommendation system would apply the similarity comparison and clustering algorithm to calculate an expected Distance Dispersion ρ_T^P for P to carpool with a particular nearby taxicab T according to six different calculation models. All calculation models are divided into the following four steps:

(1) All systems first locate a nearby taxicab set T, where taxicabs are near the origin, based on the traces of taxicabs in the dataset for a particular day.

(2) According to the manned trajectory distribution and passenger P information, we can calculate a preliminary potential destination set MD_T^P for taxicab T.

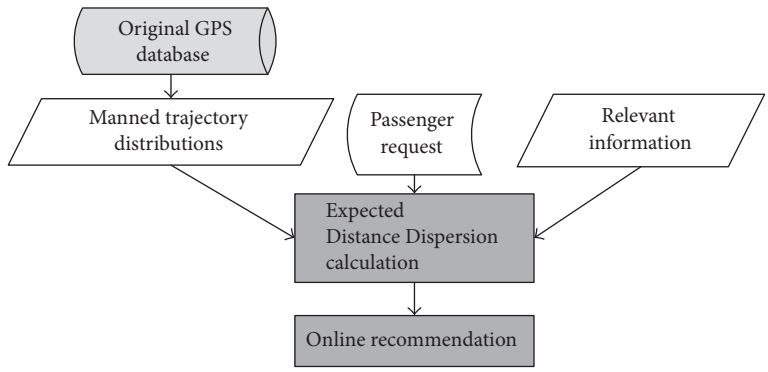


Figure 4: Main procedure.

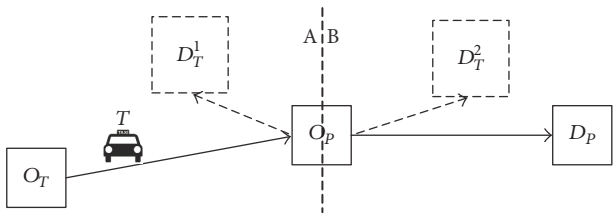


Figure 5: Basic model.

(3) Based on the context information, our system optimizes MD_T^P by removing worthless destinations to achieve a compact size in basic and advanced models. We then calculate RD_T^P containing the representatives of all potential destinations by the clustering algorithms for a further optimization.

(4) On the basis of RD_T^P, the recommendation system assigns probabilities and calculates the Distance Dispersion of this particular occupied taxi.

Basic K-Means

(1) When we receive a request, this scheme can calculate set T, where taxicabs are all near the request origin based on the real-time GPS records.

(2) By calculating the similarity between historical trajectories and the last manned trajectory of T, our system obtains the set MD_T^P, in which every potential destination has two attributes (frequency and average similarity).

(3) In the basic design, if a destination is the polar opposite of a passenger destination, then our recommendation system would eliminate this destination, due to that large consumption compared with conventional taxi service. As shown in Figure 5, when the possible destination D_T^2 of T is in B, D_T^2 is a closer destination to D_P, which diminishes consumption compared with D_T^1 in A.

K-means is then used to deeply optimize and highly generalize the characteristics of the taxi destination distribution.

(4) In the basic design, with assigning equal probabilities for the destinations in RD_T^P, *VOT* calculates a weighted average ρ_T^P by their locations.

Advanced K-Means. Advanced K-means is similar to *basic K-means* except for two differences.

In (3), the advanced design is built upon the basic design. However, in the advanced design, based on richer underlying information, our system further reduces the size of MD_T^P by two steps of depth optimization.

Step 1. We firstly census a set, called Recent Occur Destinations (ROD), which contains the destinations and their frequencies that have occurred in the recent days according to historical manned trajectories. And there are some potential destinations that do not appear in ROD or have only minimal frequencies (less than three times). Therefore, since these destinations have a small probability of being the real destination, *VOT* in the advanced model removes these destinations that have rarely occurred in recent days to improve prediction accuracy.

Step 2. If a region appears many times in a short period of time, this indicates that there has been a great service demand for this region in the last few hours. In other words, this region has a great possibility of being the real destination. Therefore, *VOT* firstly censuses these regions, which are the final destinations for manned trajectories that have occurred in recent hours. Then, *VOT* in the advanced model detects and marks the region with high frequency. At the end of clustering algorithms, we can obtain the middle region of the marked region and the cluster center in which this marked region is located. At last, the intermediate region replaces the original cluster center as the representative. These measures in Step 2 not only effectively solve the problem of short-term carpooling request surge caused by unexpected emergencies, but also compensate for the omission of real-time emergencies in Step 1.

In (4), after obtaining the clustering result from K-means, the recommendation scheme assigns probabilities to different representatives based on their individual frequencies, resulting in an accurate calculation in the Distance Dispersion. In

other words, the visits of these representatives are used as the basis for assigning probability. For example, if 10 trips starting from O_T exist in the distribution, four of them have D_T^x as their destination, whereas the others have D_T^y; our system then assigns $\Pr(D_T^x) = 4/10$ and $\Pr(D_T^y) = 6/10$ to calculate a weighted average ρ_T^P.

Basic and *advanced K-means* optimize MD_T^P by the K-means algorithm, a typical clustering algorithm based on distance. K-means uses distance as the similarity evaluation index; thus, the closer the two objects are, the greater the similarity is. The function method to find the extremum is used for the adjustment rules of iterative operation [28, 29]. The entire process is calculated as

$$\sum_{i=1}^{K}\sum_{j=1}^{N}\left(\left(F_j - F_i\right)^2 + \left(S_j - S_i\right)^2\right)^{1/2}, \tag{1}$$

where K is the number of initial cluster centers and N is the number of remaining destinations. F represents frequency, and S denotes average similarity.

The Minkowski Distance formula between two regions and the cluster center coordinate are shown below:

$$\mathrm{MK} = \left(\sum_{k=1}^{n}\left(x_{1k} - x_{2k}\right)^p\right)^{1/p} \left(\sum_{i=1}^{K+N}\frac{F_i^2}{K+N}, \sum_{i=1}^{K+N}\frac{S_i^2}{K+N}\right). \tag{2}$$

When $p = 1$, the Minkowski Distance is the Manhattan Distance; when $p = 2$, the Minkowski Distance is the Euclidean Distance.

Basic and *advanced DBSCAN* are similar to *basic* and *advanced K-means*, but they use DSBSCAN to optimize MD_T^P. DBSCAN is a spatial clustering algorithm based on density, which is not sensitive to distance. The algorithm divides the regions with sufficient density into clusters and finds the clusters of arbitrary shapes in noisy spatial databases [30, 31]. Based on the above reasons, *advanced DBSCAN* has the best performance in both Distance Dispersion and reduced total mileage on average, except when the density is uneven and the distance between clusters is very different at some time, which can also be proved in Section 6.

Basic and *advanced BIRCH* are also similar to *basic* and *advanced K-means*, but they use BIRCH to optimize MD_T^P. BIRCH is a clustering algorithm based on hierarchy [32]. This algorithm uses two concepts, namely, clustering feature and clustering feature tree, to generalize clustering description [33].

4.2.3. Online Recommendation. The algorithm recommends a real-time taxi with the minimum expected Distance Dispersion for a particular passenger by analyzing the Distance Dispersion for every nearby taxi whether in the Wander or in the Occupied Status.

5. System Implementation

5.1. Calculation Framework. Although the raw GPS dataset is typically of a large volume and interconnects multidimensional records with high resolution, much of the raw dataset is of no interest in our design. We need to map this raw physical GPS dataset to a filtered and compressed logical dataset for analysis. Moreover, we should process this raw physical GPS dataset by an intelligent method in order to meet the high timeliness and low latency requirements. In this aspect, a large data-processing framework can be a good solution to the problem of raw and massive data processing.

Spark [34] is the latest generation of software framework for distributed processing of large-scale data, which has the advantages of high efficiency, high fault tolerance, and low cost [35]. Memory distribution dataset goes into operation in Spark, which improves the performance of iterative computation by caching data in memory [36]. Thus, Spark meets the requirements of the real-time taxi recommendation system for high timeliness and low latency [37]. In conclusion, our recommendation system uses Spark to deal with the raw GPS dataset.

As a burgeoning big-data-processing model, Spark provides the basic abstraction that is a resilient distributed dataset (RDD [38]). RDD represents an immutable, partitioned collection of elements that can be operated in parallel. Data manipulation in Spark programs can be divided into three steps: the creation of RDD, the transformation of the existing RDD, and the operation of the RDD returning the computing result. In detail, before submitting the Spark program, Spark runs the program's main function and builds a Spark context. Then, Spark programs load data by abstracting data into a RDD. Finally, based on the user-defined logic, the data processing and transformation are realized on the basis of user-defined functions and the operator (map, filter, groupByKey, sortByKey, etc.) provided by Spark.

However, although the types of operators provided by Spark are rich, there are still some complex and unique operation logics, which need to be implemented by the combination with user-defined functions and the operators provided by Spark.

5.2. Historical Manned Trajectory Distribution. Each GPS record has a pair of latitude and longitude, but if the GPS latitude and longitude point are regarded as a mark of matching the trajectories, we could not map the particular trajectories because two pairs of latitude and longitude with the same value hardly exist. Therefore, we introduce regional division in *VOT*. The map is divided into many marked regions. A marked region would contain several GPS records of the same taxicabs by continuously tracking the GPS records. The taxicab trajectory can then be represented by a series of marked regions. In this manner, trajectory matching becomes possible by searching for particularly same regions.

Based on the raw large GPS dataset and regional division, *VOT* could obtain the manned trajectory distribution, in which each manned trajectory consists of a series of marked regions instead of one-by-one GPS latitude and longitude point to describe the entire taxi-manned trajectory. As shown

Table 1: Original GPS records and areas of manned trajectory.

Number	Time	Longitude	Latitude	Area	Status
23953	19:32:45 PM	114.0993	22.5451	Jd43Wd7	0
23953	19:32:49 PM	114.0989	22.5518	Jd43Wd7	1
23953	19:33:08 PM	114.0990	22.5401	Jd43Wd6	1
23953	19:33:26 PM	114.0988	22.5391	Jd43Wd6	1
⋮	⋮	⋮	⋮		⋮
23953	19:47:55 PM	114.0489	22.5321	Jd35Wd5	1
23953	19:48:01 PM	114.0479	22.5316	Jd35Wd5	1
23953	19:48:10 PM	114.0429	22.5312	Jd34Wd5	1
23953	19:51:20 PM	114.0409	22.5298	Jd34Wd5	0

(1) Input taxicabs GPS data after cleaning
(2) Using *map* transformation, the format of raw taxicabs GPS records is converted to (plate number, (date and time; marked region; status bit))
(3) Using *groupByKey* transformation, all the taxicabs GPS data of the same plate number are gathered.
(4) **if** (Using *filter* transformation, we inspect and detect if there are real-time taxicabs in Wander Status)
{
(1) Using *map* transformation, we calculate the distance between these real-time taxicabs and *P*. Then, (corresponding distance, plate number) are exported.
(2) Ascending order of corresponding distance can be obtained by *sortByKey(true)* transformation.
(3) Using *take(1)* operation, we obtain and recommend the nearest taxicab in Wander Status to the passenger *P*.
}
(5) **else**
{
(1) Using *filter* transformation, we inspect and detect if there are real-time taxicabs in Occupied Status.
(2) Using *map* transformation, GPS data for each taxicab are arranged in reverse chronological order. Then, we output plate number and the corresponding last manned trajectory, namely, several continuous GPS records in which the status bit is 1.
}

Procedure 1: Access to real-time taxicab information.

in Table 1, serval original GPS records are used as examples to demonstrate the above conversion.

The original GPS records are transformed to several marked regions (e.g., Jd43Wd6) after the regional division in Table 1. A series of raw GPS records describe the details of the above entire trajectory, which can be mapped on a given region map, corresponding to a unique carpool graph. Thus, a manned trajectory is extracted from raw GPS records and represented by a series of marked regions.

5.3. Function Implementation. The procedure of Spark data processing is a series of RDD transformations and operations. Hence, a series of key RDD transformations and operations are used to explain the critical details of the mechanisms and algorithms in this section. In the following, the significant RDD transformations and operations are described.

5.3.1. Access to Real-Time Taxicab Information. Upon receiving a request from passenger P in O_P, we first need to search for taxicabs around passenger P through the real-time GPS records. In Procedure 1, the real-time taxicabs in Wander Status or Occupied Status are obtained by testing the time and Status Bit of GPS records.

If there are several real-time taxicabs in Wander Status around P, the distances between O_P and these taxicabs are regarded as an attribute of vacant taxicab performance. Then, we select the nearest vacant taxicab to P. If there is no real-time vacant taxicab around P but only a few real-time

Step 1. Obtaining Initial Destination Sets
(1) Input the historical taxicab manned trajectory data. Then, the storage level of these trajectory data is put into *StorageLevel.MEMORY_ONLY* by *cache* method due to the need for repeated comparisons.
Input the last manned trajectory of real-time taxicabs in Occupied Status. *textFile* method is used to load the HDFS file into Spark as an initial RDD.
(2) Using *map* transformation, we can obtain the similarity between the last manned trajectory of these taxicabs and the historical manned trajectory data.
After the above operations, the new RDD with the format of (similarity, destination) is transformed.
(3) Using *sortByKey (false)* transformation, the descending order about similarity of potential destinations is obtained.
(4) Using *take(n)* operation, we can obtain n taxicab historical manned trajectories which have higher similarity, and destinations of these manned trajectories are regarded as a preliminary set MD_T^P.
(5) In order to deal with these data more conveniently and quickly, we change the form of MD_T^P to (destination, similarity) by *map* transformation. After that, the new MD_T^P is exported to HDFS to facilitate filtering operations later.

Step 2. Forecast Final Destinations
(1) *textFile* method loads and abstracts MD_T^P into RDD, and then *VOT* gathers the similarity of the same potential destination by *groupByKey* transformation.
(2) Using multiple operators provided by Spark and user-defined functions, downsized and optimized MD_T^P is obtained in basic and advanced models.
(3) Using *foreach* operation, we calculate the visit frequency and the average similarity of potential destinations in MD_T^P and export the data in the format of (potential destinations, (frequency, average similarity)) to HDFS.
(4) The new MD_T^P in HDFS is abstracted as RDD by the *textFile* method. Then, through a series of transformations and actions including user-defined functions, we implement and complete three different types of clustering algorithms and output the representatives RD_T^P of MD_T^P.
The format of initial RD_T^P is ((destinations and these attributes in Cluster A), (destinations and these attributes in Cluster B)...)
(5) Based on *map* transformation and initial RD_T^P, cluster centers and total visit frequency of clusters are calculated by user-defined functions.
The format of the output file is ((the cluster center C_A and total visit frequency N_A of Cluster A),...).
(6) We traverse each element of the RDD by the *foreach* operation to count the total visit frequency N. Then, the ultimate RD_T^P with the format of $((C_A, N_A/N), \ldots)$ is exported to HDFS.

Procedure 2

taxicabs in Occupied Status, we further calculate the last manned trajectory of the real-time taxicabs in Occupied Status (see Procedure 1).

5.3.2. Potential Destinations of Occupied Taxicabs. In Procedure 2, in order to obtain potential destinations of real-time taxicabs in Occupied Status, our algorithm is roughly divided into two steps.

Step 1 (obtaining initial destination sets MD_T^P). By the comparison between the last manned trajectory of these real-time occupied taxicabs and the historical manned trajectory data, *VOT* calculates and acquires the destinations of n trajectories which have higher similarity, namely, MD_T^P.

Step 2 (forecast final destinations RD_T^P). Based on the frequency and average similarity of every potential destination in MD_T^P, different clustering algorithms (K-means, density-based spatial clustering of applications with noise (DBSCAN), and balanced iterative reducing and clustering using hierarchies (BIRCH)) complete clustering operations. Then, we calculate and regard the cluster centers set RD_T^P as the representative of potential destinations in the same cluster.

5.3.3. Distance Dispersion Calculation and Optimal Recommendation. In order to screen out the real-time occupied taxicab with the best carpooling performance, Procedure 3 is divided into two steps.

Step 1. Our algorithm calculates the Distance Dispersion of every real-time taxicab in Occupied Status.

Step 2. This real-time taxicab in Occupied Status with the best carpooling performance is selected by *VOT* and recommended to the particular passenger P.

As shown in Procedure 3, our recommendation strategy specifies a *map* transformation that puts the representatives RD_T^P and P's request (origin and destination) as input file to calculate Distance Dispersion of real-time occupied taxicabs. The generic calculation formulas are as follows:

$$\rho_T^P = \sum_{D_T \in RD_T^P} \Pr(D_T) \left(\frac{\mathrm{EM}_{D_T}^{D_P} + \mathrm{MH}_{D_T}^{D_P}}{2} \right)$$

$$\mathrm{EM} = \left(\sum_{k=1}^{n} (x_{1k} - x_{2k})^2 \right)^{1/2} \quad (3)$$

$$\mathrm{MH} = \sum_{k=1}^{N} |x_{1k} - x_{2k}|,$$

(1) Input and abstract clustering results RD_T^P to new RDD.
(2) Using *map* transformation, the carpool performance of a single potential destination of a real-time taxicab in Occupied Status is quantified (Distance Dispersion) by user-defined functions.
(3) Using *reduceByKey* transformation, the carpool performance of a single real-time occupied taxicab is obtained by aggregating all Distance Dispersion of potential destinations in RD_T^P.
(4) By the *sortByKey* transformation, the descending order of the carpool performance (the ascending order of Distance Dispersion) is processed.
(5) Using *take(1)* operation, we obtain and recommend the real-time taxicab in Occupied Status with the best carpooling performance to the passenger P.

Procedure 3: Distance Dispersion calculation and optimal recommendation.

where RD_T^P is the representative of MD_T^P and D_T is a representative of the potential destinations. $\mathrm{EM}_{D_T}^{D_P}$ is the Euclidean Distance between passenger P's destination D_P and the real-time taxicab T's destination D_T. $\mathrm{MH}_{D_T}^{D_P}$ is the Manhattan Distance between these destinations. Every destination has a different probability according to the frequency by which it appears in RD_T^P. $\Pr(D_T) = |D_T|/|RD_T^P|$, where $|D_T|$ is the total frequencies of D_T and $|RD_T^P|$ is the total frequencies of all destinations. If T is a vacant taxicab, then operations return 0 as the Distance Dispersion, given that no distance exists for a vacant taxicab.

6. Evaluation

The sample dataset, which contains 4.5 million GPS raw records of 14747 taxicabs, is used to test our recommendation system. Owing to the large size of the dataset, we find a major amount of errant records. Two main errors exist: (i) abnormal error (e.g., although the state value is 1, which means the taxicab is moving, the continuous GPS records show that the latitude and longitude are maintained, which is illogical) and (ii) matching error (after matching with the electronic map, the GPS coordinates indicate that a taxicab is off the road) [39].

These errors may result from different causes, such as GPS device malfunctions, software issues, and human factors. Before data processing, we clean the original data using simple preprocessing operations to delete abnormal and invalid GPS records.

6.1. Evaluation Setup. In this study, *VOT* compares three clustering algorithms (K-means, DBSCAN, and BIRCH) in basic and advanced models. The taxi-manned trajectory distributions, which show real passenger requests, can be obtained based on the historical GPS datasets. Real requests, which occurred in the dataset at one day, are regarded as future requests to test our recommendation system. Based on a specific manned trajectory, for example, take-in time T_x, origin area O_x, drop-off time T_y, and destination area D_y, in the taxi-manned trajectory distributions, a passenger request (request time T_x, origin O_x, and destination D_y) can be generated.

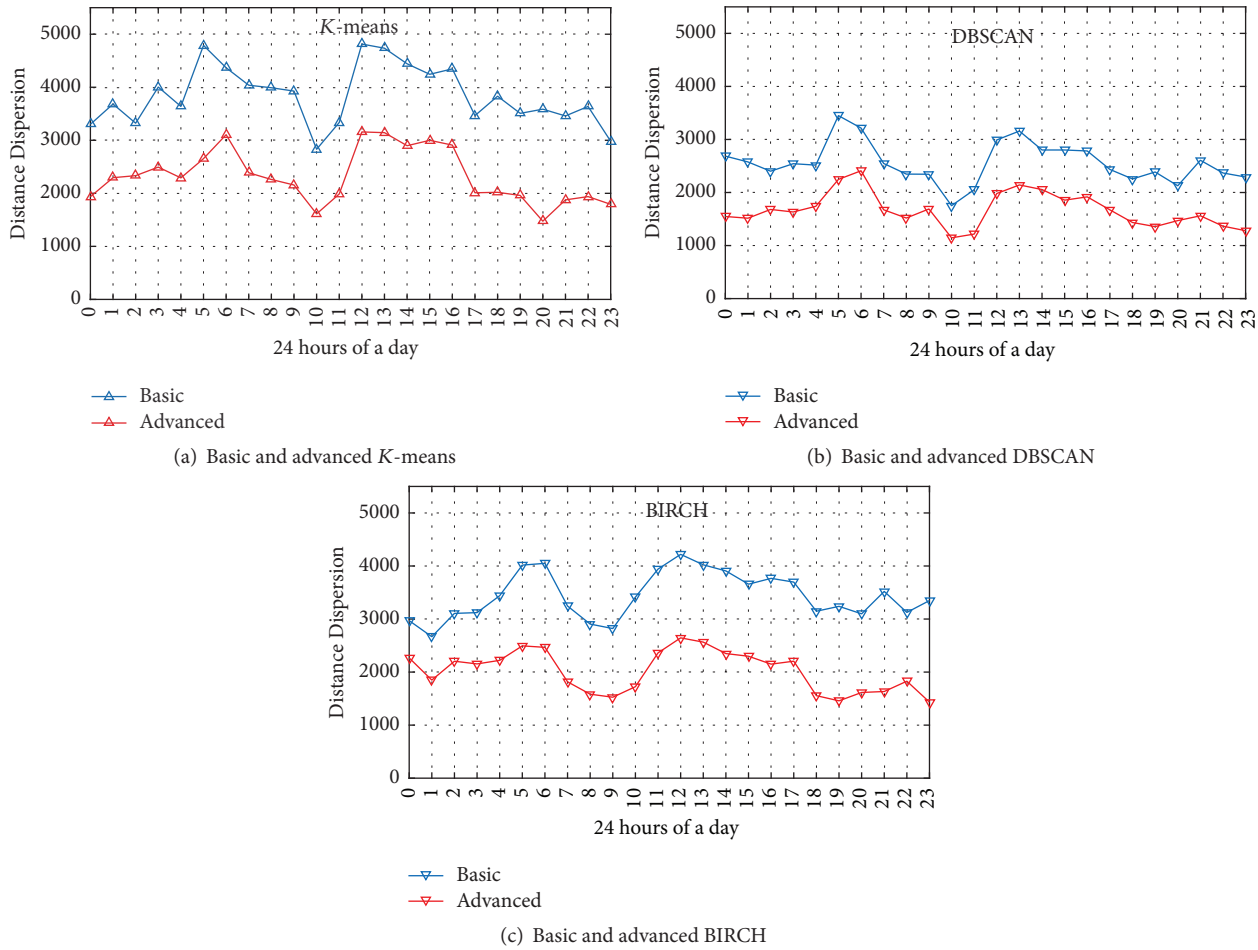
All recommendation algorithms match this actual request with the real-time GPS records for a nearby taxicab set T based on the trajectories of taxicabs in the dataset for a particular day. If vacant taxicabs exist in T, all recommendation algorithms suggest the closest vacant taxicab to passengers. Otherwise, basic K-means calculates the Distance Dispersion for every occupied taxicab in T based on the basic model and the K-means algorithm in Section 4.2 and then recommends the occupied taxicab with the minimum attribute value. Other algorithms function similarly, except that these algorithms calculate Distance Dispersion based on different clustering algorithms (DBSCAN and BIRCH) and different models (advanced models).

Distance Dispersion is regarded as a key metric to show the efficiency of taxicab service, which is obtained by $(\mathrm{EM}_{D_T}^{D_P} + \mathrm{MH}_{D_T}^{D_P})/2$; this metric is used to evaluate the closeness between passenger and taxicab destinations. For vacant taxicabs, the Distance Dispersion is 0; for occupied taxicabs, we compare and recommend the occupied taxicab with the minimum Distance Dispersion to passengers. Hence, Distance Dispersion can provide a recommendation which maximizes passengers' interests for both carpooling and conventional taxicab services.

What is more, we justify carpooling services by showing reduced total mileage (%). Unlike Distance Dispersion which concentrates on the interests of an individual passenger, reduced total mileage is used to calculate how much total mileage can be reduced (leading to less gas exhaust emissions and less traffic congestion) by an efficient system recommending more suitable taxicabs in Occupied Status for passengers. Supposing M is the total mileage for individually delivering all passengers and m is the total mileage for delivering all passengers with either conventional taxi or carpool service, then the percentage of reduced mileage equals $(M - m)/M$.

In order to prove the superiority of Distance Dispersion, we use actual detour ratio to evaluate *VOT*, which is regarded as a key metric to show the efficiency in other recommendation systems. Compared to conventional taxi service, carpooling service has a detour distance (ActualDistance − DirectDistance). Thus, actual detour ratio can be obtained by (ActualDistance − DirectDistance)/DirectDistance.

Then, we propose a new parameter, called Real Prophecy Distance (RPD), to demonstrate the ratio of correctly predicted destinations, which is obtained by quantifying this distance between true destinations and forecasted cluster centers.



(a) Basic and advanced K-means

(b) Basic and advanced DBSCAN

(c) Basic and advanced BIRCH

Figure 6: Distance Dispersion (M).

We evaluate *VOT* at different cluster numbers and various region sizes, according to the above metrics. This evaluation leads to different service effects in terms of the same algorithm. The default setting of cluster number is 5, and the default setting of region length is 600 M. For the entire dataset, we use the real requests from a one-day dataset and test all the algorithms with the trajectories of taxicabs on other days. The average results are reported.

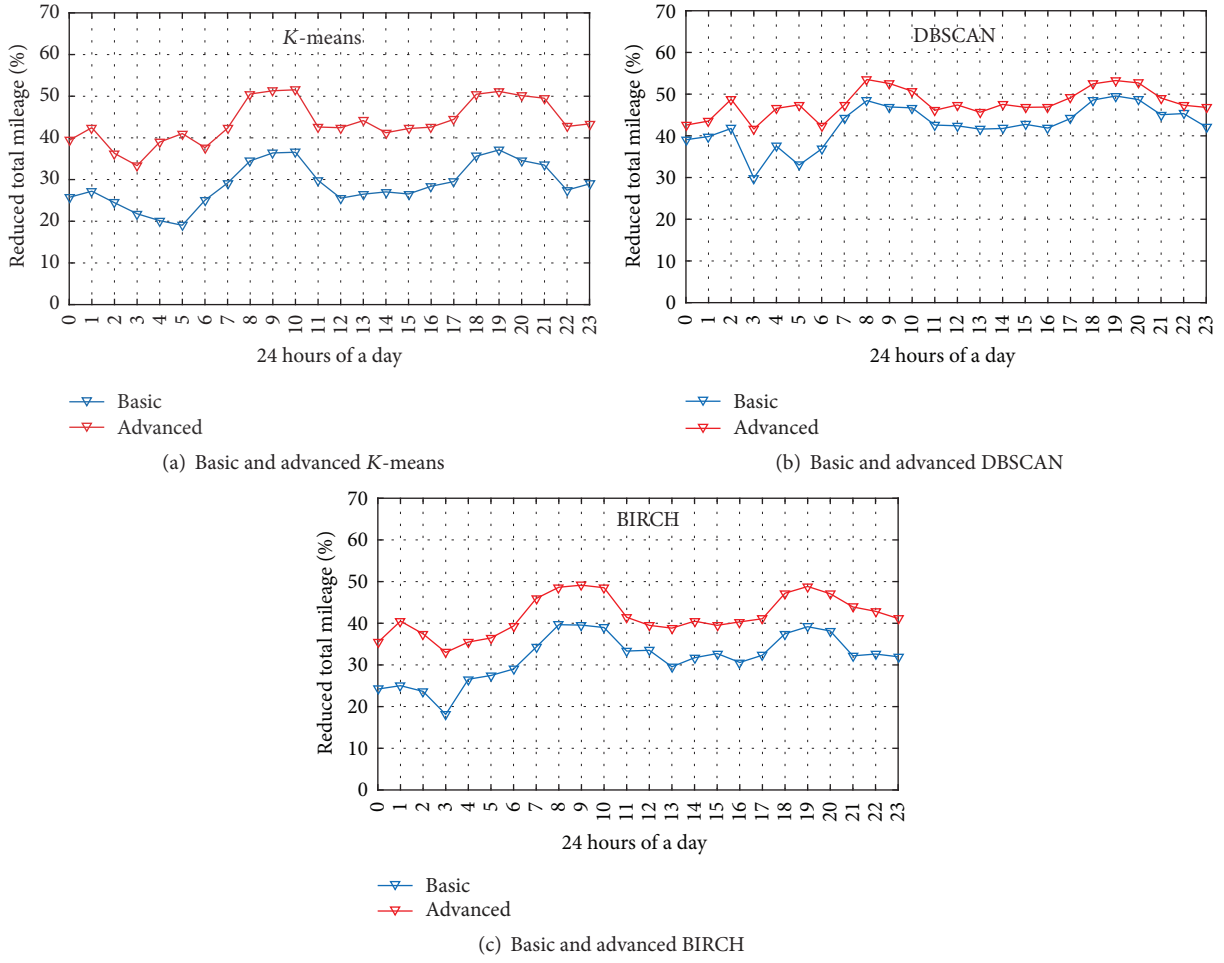
6.2. Distance Dispersion. In this subsection, we investigate the average Distance Dispersion performance.

Figure 6 shows the average Distance Dispersion in different 1 h time slots of one day. During rush hours, such as 8:00 to 10:00 AM and 18:00 to 20:00 PM, the average Distance Dispersion for all versions is lower than during nonrush hours, such as 1:00 to 7:00 AM. This result is due to the fact that passengers during rush hours have more fixed destinations and that more historical GPS data are available for predictions. Therefore, our recommendation system can more accurately predict the destinations of occupied taxicabs by context information and manned trajectory distributions.

A comparison of the three clustering algorithms indicates that DBSCAN has the best performance, with a minimum Distance Dispersion, and performs well in both basic (2.560 km) and advanced (1.671 km) scenarios, which effectively guarantees the interests of passengers. That is because DBSCAN can find clusters of arbitrary shapes, which provides it with the highest prediction accuracy. K-means has a good carpool quality in the advanced model, but the performance is poor in the basic model, with a large difference at 1.524 km. That is because many abnormal and worthless data seriously interfere with K-means in the basic model.

6.3. Reduced Total Mileage. In this subsection, we evaluate the performance of *VOT* through the percentage of reduced total mileage (%).

Figure 7 shows the percentage of reduced total mileage in different 1 h time slots. During rush hours, such as 8:00 to 10:00 AM and 18:00 to 20:00 PM, the percentages of reduced total mileage for all six schemes are higher than those on nonrush hours, especially 1:00 to 7:00 AM. This result is attributed to the increased carpooling service demands during rush hours compared to those on nonrush hours. Meanwhile, with more accurate carpooling recommendations for passengers, our recommendation system also leads to a much bigger



(a) Basic and advanced K-means

(b) Basic and advanced DBSCAN

(c) Basic and advanced BIRCH

FIGURE 7: Reduced total mileage (%).

reduction in the total mileage to deliver the same number of passengers than the reduced total mileage on nonrush hours.

In both K-means and BIRCH algorithms, the advanced model outperforms the basic one by 15.06% and 10.05% on average, respectively, indicating the superiority of the advanced model. With high carpool quality, DBSCAN is not sensitive to basic and advanced scenarios, which confirms our previous observations. From the overall view, DBSCAN is the best choice because of its stable and high carpool quality with 47.84% in reduced total mileage on average in the advanced model. Nevertheless, K-means outperforms DBSCAN at some hours in the advanced model, such as 9:00-10:00 and 21:00.

6.4. Actual Detour Ratio. Figure 8 shows the performance for the average actual detour ratio in different 1 h time slots of one day. During the busy commuting time, such as 8:00 to 10:00 AM and 18:00 to 20:00 PM, the average actual detour ratio for all three algorithms in the advanced model is higher than those on nonbusy hours, such as 1:00 to 7:00 AM. The variation trend of *VOT* is almost the same as that of similar researches.

With the best performance among the three algorithms, the actual detour ratio (%) of DBSCAN at any 1 h time slots of one day is no more than 10%, which is clearly superior to other researches in the busy commuting time. Then, although K-means and BIRCH do not have a good performance, their worst cases are still no more than 15%,

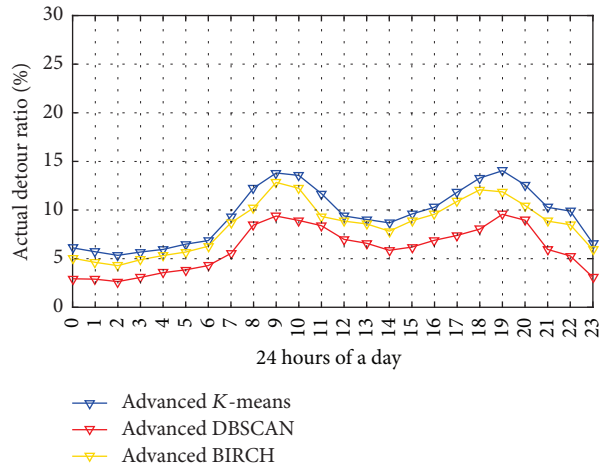
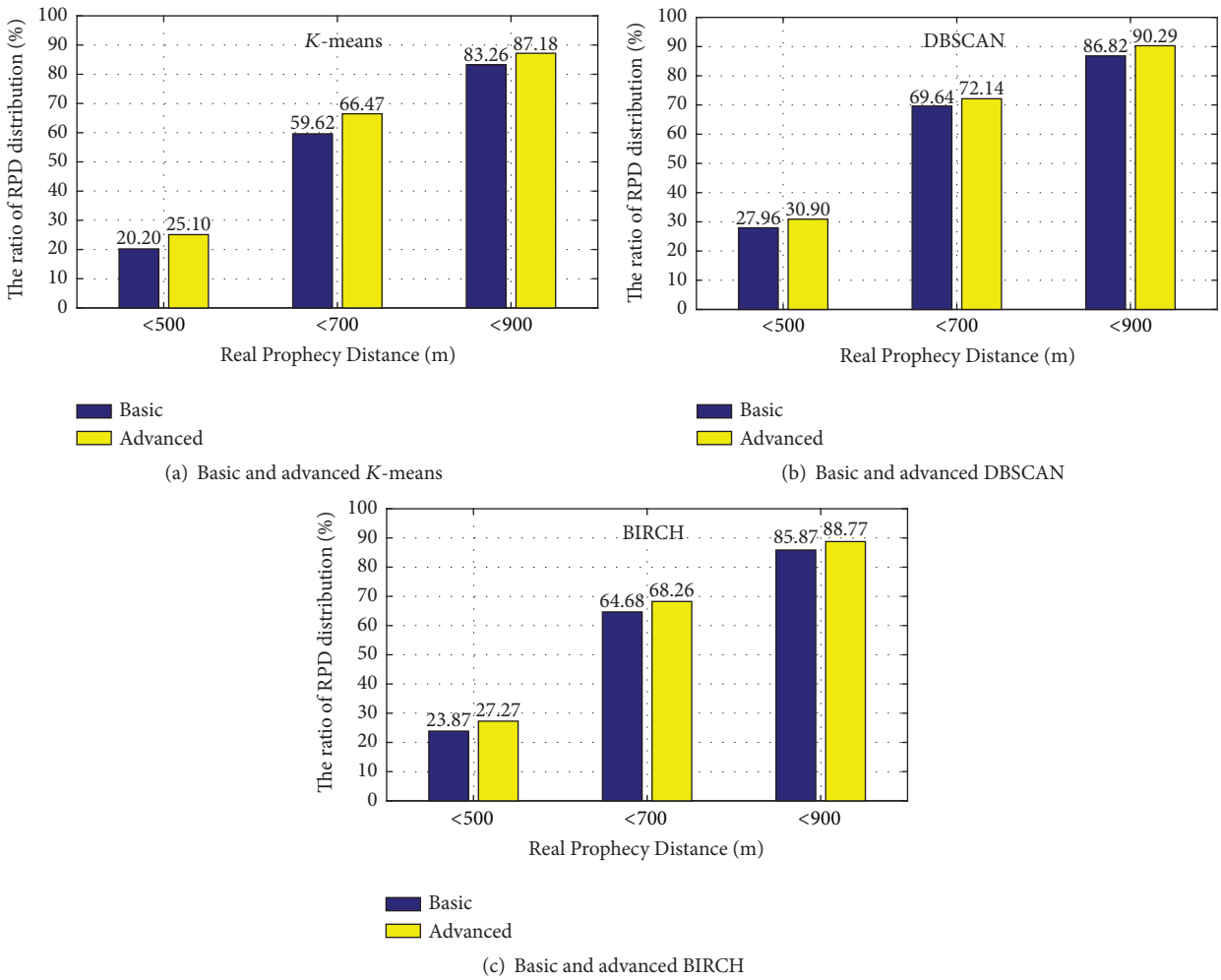


FIGURE 8: Actual detour ratio (%).



(a) Basic and advanced *K*-means

(b) Basic and advanced DBSCAN

(c) Basic and advanced BIRCH

Figure 9: Real Prophecy Distance (M).

14.04%, and 12.84% respectively. What is more, there is only 3.48% difference on average between advanced DBSCAN with the best performance and advanced *K*-means with the worst performance. In other words, all versions of *VOT* in the advanced model can fully guarantee and control the actual detour ratio.

Based on the results of the above supplementary experiments, we demonstrate that *VOT* can perform well in both the interests of passengers (actual detour ratio (%)) and the mitigation of gas exhaust emissions (reduced total mileage (%)). Therefore, Distance Dispersion is regarded as a key metric to show the efficiency of conventional and carpooling service in *VOT*, instead of actual detour ratio (%).

6.5. Real Prophecy Distance Distribution. Figure 9 shows the percentage of Real Prophecy Distance distribution under the default region length (600 M).

The distributions (<900 M) of RPD for six versions are all over 85% (except for basic *K*-means, 83.26%), especially advanced DBSCAN with 90.29%. Remarkably, because the default region length is set to 600 M, the worst condition of the RPD distributions (<900 M) is that there is only less than two regions between true destinations and forecasted cluster centers. What is more, the distributions (<500 M) of RPD for six versions are almost all over 25%, in which advanced DBSCAN has the best performance with 30.90%. Notably, RPD, which is less than 500 M, means only one situation: the predicted cluster centers are in the same region as the real destinations (or adjacent when region length is 400 M). In other words, the prediction result must be absolutely correct, if RPD is less than 500 M.

For *K*-means, DBSCAN, and BIRCH, the advanced model outperforms the basic model by 5.22%, 2.97%, and 3.29% on average, clearly indicating the superiority of the advanced model. A comparison of these clustering algorithms suggests that DBSCAN has the best performance. And DBSCAN works well in the three different RPD distributions (<500 M (30.90%), <700 M (72.14%), and <900 M (90.29%)), which clearly demonstrates the prediction accuracy of *VOT*. These results confirm that *VOT* is actually able to guarantee high prediction accuracy.

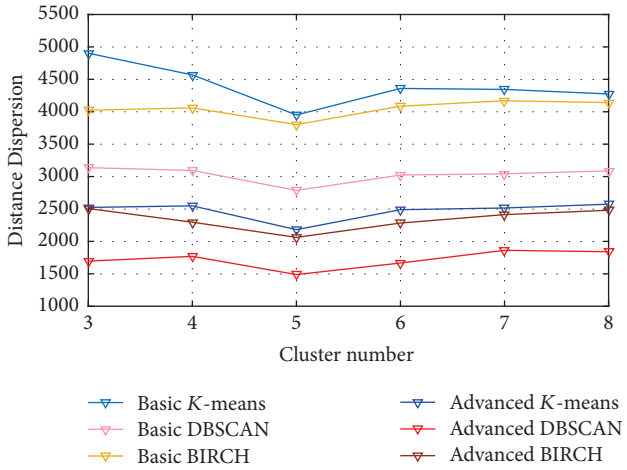


Figure 10: Distance Dispersion versus cluster number.

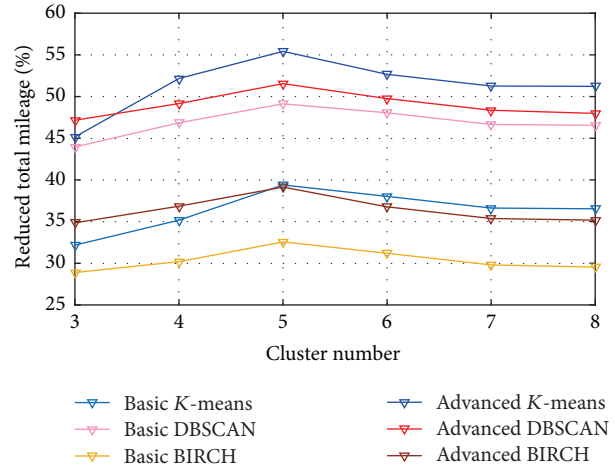


Figure 11: Reduced total mileage (%) versus cluster number.

6.6. Cluster Number Effect. In this subsection, we learn the influence of recommendation radius on *VOT* performance at 9:00 AM of one day.

6.6.1. Distance Dispersion with Different Cluster Numbers. Figure 10 shows the effect of different cluster numbers on the performance of the 6 schemes in terms of Distance Dispersion. We change the cluster number from 3 to 8, which in turn alters the number of destinations to be used to summarize the distribution characteristics of occupied taxicabs.

For all six visions of *VOT*, the Distance Dispersion under the advanced model is invariably better than that in the basic model. That is because better recommendations are provided to passengers by eliminating the worthless candidate destinations in the former. Minimum Distance Dispersion is achieved when the cluster number is 5, and the increase for 6 versions of *VOT* slows down when the cluster number is close to 8. Compared with the numbers 3 and 8 that cannot precisely generalize the characteristics of destination distribution, the number 5 is consistent with the destination distribution of a vast majority of taxicabs.

6.6.2. Reduced Total Mileage (%) with Different Cluster Numbers. Figure 11 shows the effects of different cluster numbers on the percentage of reduced total mileage at 9:00 AM of one day.

The maximum reduced total mileage occurs when the number of the clusters is 5. When the cluster number is close to 8, the decrease for 6 versions of *VOT* slows down. In other words, the minimum Distance Dispersion and the maximum reduced total mileage, which indicate the best carpool quality, occur at 5 at the same time. Thus, we recommend that the number of clusters be set to 5 for enhanced carpool quality. And in the advanced model, K-means outperforms DBSCAN in terms of reduced total mileage at 9:00 of one day, which confirms our previous observations.

6.7. Region Length Effect. In this subsection, we study the effect of recommendation radius on *VOT* performance for 24 h on one day in the advanced model. Due to the great similarity in tendency of the three algorithms, we just present the performance in K-means algorithm.

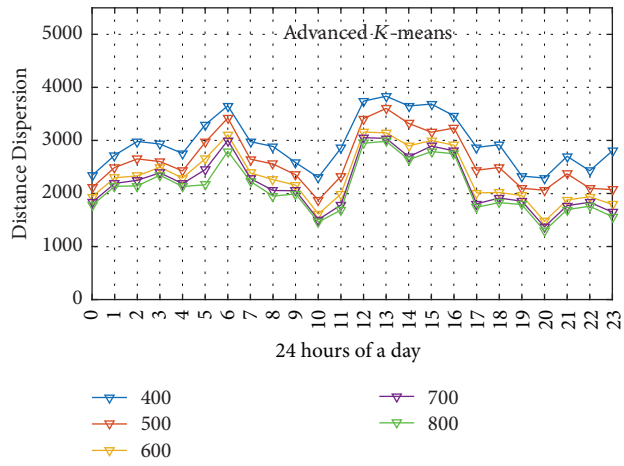


Figure 12: Distance Dispersion in advanced K-means versus region length.

6.7.1. Distance Dispersion with Different Region Lengths. Figure 12 shows the effect of different region lengths in advanced K-means on Distance Dispersion. We change the region length from 400 M to 800 M, which increases the size of potential taxicabs that can be recommended and the number of similar manned trajectories that can be analyzed.

For K-means, with the increase in the radius from 400 M to 800 M, the performance of *VOT* decreases. Nonetheless, the decrease slows down when the region length is close to 800 M, which is due to the fact that the radius is large enough to have a sufficient number of similar taxicab-manned trajectories and taxicabs for analysis and inference, and an even larger radius would not help. DBSCAN and BIRCH also have the same trend. But there are still different trends for K-means, DBSCAN, and BIRCH between 400 M

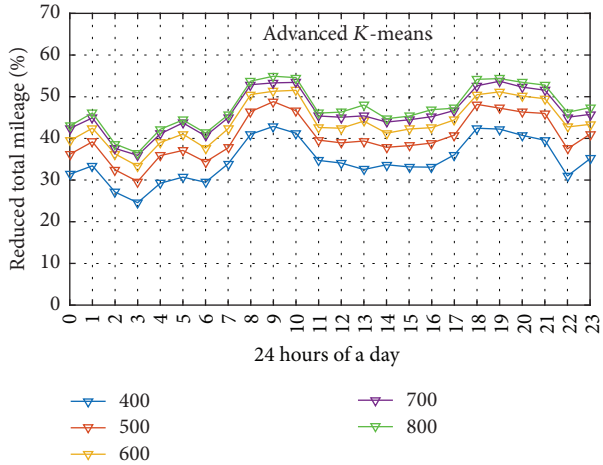


Figure 13: Reduced total mileage (%) in advanced K-means versus region length.

and 800 M, that is, 850.0760 M, 491.1766 M, and 671.8267 M, respectively.

Similar trends are maintained when the radius increases from 400 M to 800 M, such as a better performance from 18:00 to 20:00 and a worse performance from 1:00 to 7:00 AM, which verify the previous inference in the previous sections.

6.7.2. Reduced Total Mileage with Different Region Lengths. Figure 13 shows the effects of different region lengths on the percentage of reduced total mileage for 24 h on one day.

With the increase in the radius from 400 M to 800 M, the reduced total mileage of K-means in the advanced model increases given the increased carpooling service demands and the more accurate inference available. However, the increase for K-means slows down when the region length is close to 800 M. Hence, the default region length is set to 600 M because the radius is sufficiently large to provide accurate inference and calculation (only 2.76% between 600 M and 800 M), and an even larger radius is not unnecessary. DBSCAN and BIRCH also have the same trends.

The increase in region length from 400 M to 800 M leads to the largest difference in the performance of K-means between 400 M and 800 M, that is, 12.53%. By contrast, the difference in the DBSCAN performance is insignificant (i.e., 5.80%) because K-means (based on distance) is sensitive to the change in region length, whereas DBSCAN (based on density) is unresponsive to this change. Compared with K-means, the performance of BIRCH has only 9.17% increasement when the region length varies from 400 M to 800 M.

6.7.3. Real Prophecy Distance Distribution with Different Region Lengths. In this section, we evaluate the influence of region length on Real Prophecy Distance distribution under the advanced model.

Tables 2, 3, and 4 show the effect of different region lengths on the three different RPD distributions (<500 M, <700 M, and <900 M) in the advanced model.

For the three different clustering algorithms, with the increase in the region length from 400 M to 600 M, the ratio

Table 2: Real Prophecy Distance <500 M versus region length.

<500	400	500	600	Max. difference
K-means	24.0256	24.6381	25.0975	1.0719
DBSCAN	30.6759	30.7728	30.8996	0.2237
BIRCH	26.2619	26.8249	27.2711	1.0092

of RPD shows an increasing tendency. That is because a larger region length enlarges the range of a single grid, which increases the possibility that the inferred cluster centers contain the GPS records of the real destinations. But for the three different RPD distributions (<500 M, <700 M, and <900 M), the performance under 600 M outperforms that under 400 M by only 0.768%, 3.334%, and 4.740% on average. Specifically, the minimal variation tendency of the RPD distributions (<500 M) is 1.07%, 0.22%, and 1.01% for K-means, DBSCAN, and BIRCH, respectively. In other words, even if the region length is set to 400 M, all versions of *VOT* in the advanced model also can guarantee good prediction accuracy for the three RPD distributions.

In addition, in contrast to our previous comparison experiments from 400 M to 800 M in the initial manuscript, we do not carry out experiments with region lengths of 700 M and 800 M. This is because if the region length is too long, the situation satisfying the RPD distribution tends to be homogeneous. For example, when the region length is set to 800 M, the distributions (<500 M and <700 M) of RPD are quite consistent. This results in an obscure tendency of RPD distribution. Therefore, these inconclusive experiments are not executed in this section.

7. Discussion

Although *VOT* provides good carpooling performance, there is room for further enhancements. Discussed below is the system feasibility or implementability that warrants further investigation.

7.1. Changes in Existing Taxicab System. Although there is no need to build a completely new taxicab network, further optimization and promotion are necessary to the existing taxicab system for a better service. For example, a convenient two-way communication needs to be deployed between the taxicabs and the backend server, instead of one-way communication via GPS. With the development and popularization of the fourth-generation mobile communication technology, the convenience and practicability of mobile devices provide an opportunity for realization of two-way communication. Thus, we will study this respect in the further work.

7.2. An Acceptance by Passengers of Sharing the Taxi. In *VOT*, we can only realize whether the taxicab has passengers via the Status Bit of the GPS records. But if the two-way communication between the taxicabs and the backend server is realized successfully, the number of existing passengers in real-time taxicabs can be obtained by uploading the passengers' information.

Table 3: Real Prophecy Distance <700 M versus region length.

<700	400	500	600	Max. difference
K-means	61.4596	64.0293	66.4653	5.0057
DBSCAN	71.0951	71.5964	72.1402	1.0451
BIRCH	64.3057	66.4358	68.2569	3.9512

Table 4: Real Prophecy Distance <900 M versus region length.

<900	400	500	600	Max. difference
K-means	80.0751	83.6194	87.1817	7.1066
DBSCAN	89.1928	89.7317	90.2919	1.0991
BIRCH	82.7529	85.5416	88.7672	6.0143

Then, *VOT* can provide personalized carpooling options according to the preferences of passengers. For example, the acceptable number of taxi-sharing passengers is two and female only; two and male only; two and no request for male or female preference; three and female only; three and male only; three and no request for male or female preference; no request. We believe that a variety of carpool preferences options can provide passengers with more comfortable carpooling services.

7.3. The Support from Relevant Law. Through the careful and extensive investigation, currently in China, voluntary carpooling is legally a contractual relationship that belongs to the agreement of the parties' autonomy. The drivers have the obligations for ensuring the passenger safety. If man-made accidents or unforeseen events happen, the accidents should be dealt with based on the "General Principles of Civil Law" [40], "Law of Tort Liability" [41], and "Road Traffic Safety Law" [42].

There are currently no specific laws and regulations to restrict taxicab carpooling services. With the popularity of the concept of vehicle sharing, the government and a large number of researchers are actively promoting the introduction of relevant laws.

7.4. The Extra Benefit in Fleet Managers. Because reduced total mileage (%) can reach 47.84%, the cost for delivering all passengers could be significantly reduced. Namely, taxis can accomplish more delivery tasks at the same fuel costs. This could increase the income of the company and the drivers. And there are some researches [43–46] about the benefit for passengers. In the further work, the benefit for the fleet managers and passengers will be increased as an important consideration in advanced *VOT*.

8. Conclusion

In this work, we analyze, design, and evaluate a recommendation system for both carpooling and regular taxi services based on large-scale historical GPS records. Our recommendation system mines taxi-manned trajectory distributions from a historical GPS dataset. Real requests are extracted from taxi-manned trajectory distributions, and either a taxi in Wander Status with no Distance Dispersion or an occupied taxi with minimal Distance Dispersion is recommended to particular passengers. We employ a generic big-data-processing model, Spark, to efficiently handle the raw GPS dataset. Using the real-world dataset containing 14747 taxi GPS records to evaluate the system, the ratio of range (between forecasted and actual destinations) of less than 900 M can reach 90.29%, which effectively guarantees the interests of passengers. Our recommendation system can significantly reduce the total mileage (47.84% on average). Nearly half of the total mileage of the taxi is reduced, thereby effectively reducing the air and soil pollution. Meanwhile, the average reduced total mileage of 0:00 to 7:00 is increased to 45.03%, which outperforms other systems by 35%. For actual detour ratio, *VOT* and others have similar performances, even better in rush hours.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities under Grant no. 2017QNA20.

References

[1] S. Burgaz, G. Cakmak Demircigil, B. Karahalil, and A. E. Karakaya, "Chromosomal damage in peripheral blood lymphocytes of traffic policemen and taxi drivers exposed to urban air pollution," *Chemosphere*, vol. 47, no. 1, pp. 57–64, 2002.

[2] C. M. Lytle, B. N. Smith, and C. Z. McKinnon, "Manganese accumulation along Utah roadways: a possible indication of motor vehicle exhaust pollution," *Science of the Total Environment*, vol. 162, no. 2-3, pp. 105–109, 1995.

[3] G. A. Rhys-Tyler, W. Legassick, and M. C. Bell, "The significance of vehicle emissions standards for levels of exhaust pollution from light vehicles in an urban area," *Atmospheric Environment*, vol. 45, no. 19, pp. 3286–3293, 2011.

[4] J.-C. Weng, Y.-Q. Zhai, X.-J. Zhao, and J. Rong, "Floating car data based taxi operation characteristics analysis in beijing," in *Proceedings of the WRI World Congress on Computer Science and Information Engineering (CSIE '09)*, pp. 508–512, April 2009.

[5] J. Hao, J. Hu, and L. Fu, "Controlling vehicular emissions in Beijing during the last decade," *Transportation Research Part A: Policy and Practice*, vol. 40, no. 8, pp. 639–651, 2006.

[6] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong, "A cost-effective recommender system for taxi drivers," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*, pp. 45–54, August 2014.

[7] L. Tang, X. Chang, and Q. Li, "The knowledge modeling and route planning based on taxi' experience," *Acta Geodaetica et Cartographica Sinica*, vol. 39, no. 4, pp. 404–409, 2010.

[8] R. Wolfler Calvo, F. de Luigi, P. Haastrup, and V. Maniezzo, "A distributed geographic information system for the daily car pooling problem," *Computers and Operations Research*, vol. 31, no. 13, pp. 2263–2278, 2004.

[9] C.-C. Tao, "Dynamic taxi-sharing service using intelligent transportation system technologies," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07)*, pp. 3204–3207, September 2007.

[10] R. Baldacci, V. Maniezzo, and A. Mingozzi, "An exact method for the car pooling problem based on Lagrangean column generation," *Operations Research*, vol. 52, no. 3, pp. 422–439, 2004.

[11] Y. Huang, F. Bastani, R. Jin, and X. S. Wang, "Large scale real-time ridesharing with service guarantee on road networks," in *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06)*, pp. 2017–2028, September 2006.

[12] S. Ma, Y. Zheng, and O. Wolfson, "T-share: a large-scale dynamic taxi ridesharing service," in *Proceedings of the 29th IEEE International Conference on Data Engineering (ICDE '13)*, pp. 410–421, IEEE, Brisbane, Australia, April 2013.

[13] Y. Fu, Y. Fang, C. Jiang, and J. Cheng, "Dynamic ride sharing community service on traffic information grid," in *Proceedings of the International Conference on Intelligent Computation Technology and Automation (ICICTA '08)*, pp. 348–352, October 2008.

[14] A. Attanasio, J.-F. Cordeau, G. Ghiani, and G. Laporte, "Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem," *Parallel Computing*, vol. 30, no. 3, pp. 377–387, 2004.

[15] P. Healy and R. Moll, "A new extension of local search applied to the Dial-A-Ride Problem," *European Journal of Operational Research*, vol. 83, no. 1, pp. 83–104, 1995.

[16] J.-F. Cordeau, "A branch-and-cut algorithm for the dial-a-ride problem," *Operations Research*, vol. 54, no. 3, pp. 573–586, 2006.

[17] L. M. Hvattum, A. Løkketangen, and G. Laporte, "A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems," *Networks*, vol. 49, no. 4, pp. 330–340, 2007.

[18] R. Bajaj, S. Ranaweera, and D. Agrawal, "GPS: location-tracking technology," *Computer*, vol. 35, no. 3, pp. 92–94.

[19] D. Zhang, T. He, Y. Liu, and J. A. Stankovic, "CallCab: A unified recommendation system for carpooling and regular taxicab services," in *Proceedings of the IEEE International Conference on Big Data (Big Data '13)*, pp. 439–447, October 2013.

[20] D. Zhang, T. He, Y. Liu, S. Lin, and J. A. Stankovic, "A carpooling recommendation system for taxicab services," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 254–266, 2014.

[21] H. Tian, "Data Description for UrbanCPS [EB/OL]," http://www-users.cs.umn.edu/~tianhe/BIGDATA/.

[22] Z. Gui, Y. Xiang, and Y. Li, "Parallel discovering of city hot spot based on taxi trajectories," *Huazhong Keji Daxue Xuebao (Ziran Kexue Ban)/Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 40, no. 1, pp. 187–190, 2012.

[23] X. Li, G. Pan, Z. Wu et al., "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.

[24] S. Liu, Y. Liu, L. M. Ni, J. Fan, and M. Li, "Towards mobility-based clustering," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pp. 919–927, July 2010.

[25] Q. Niu, T. Huan, and P. Chen, "NMCT: a novel Monte Carlo-based tracking algorithm using potential proximity information," *International Journal of Distributed Sensor Networks*, vol. 2016, Article ID 7061486, 10 pages, 2016.

[26] D. Zhang, T. He, S. Lin, S. Munir, and J. A. Stankovic, "Online Cruising Mile Reduction in Large-Scale Taxicab Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 11, pp. 3122–3135, 2015.

[27] D. Agrawal, P. Bernstein, E. Bertino et al., "Challenges and opportunities with big data," 2012, http://www.cra.org/ccc/files/docs/init/bigdatawhitepaper.pdf.

[28] W. Zhao, H. Ma, and Q. He, "Parallel K-means clustering based on mapreduce," in *Cloud Computing*, vol. 5931 of *Lecture Notes in Computer Science*, pp. 674–679, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[29] S. Gopalani and R. Arora, "Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means," *International Journal of Computer Applications*, vol. 113, no. 1, pp. 8–11, 2015.

[30] D. Han, A. Agrawal, W.-K. Liao, and A. Choudhary, "A novel scalable DBSCAN algorithm with spark," in *Proceedings of the 30th IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW '16)*, pp. 1393–1402, May 2016.

[31] B.-R. Dai and I.-C. Lin, "Efficient map/reduce-based DBSCAN algorithm with optimized data partition," in *Proceedings of the IEEE 5th International Conference on Cloud Computing (CLOUD '12)*, pp. 59–66, June 2012.

[32] T. Sun, C. Shut, F. Li, H. Yu, L. Ma, and Y. Fang, "An efficient hierarchical clustering method for large datasets with map-reduce," in *Proceedings of the International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '09)*, pp. 494–499, December 2009.

[33] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, vol. 25, no. 2, pp. 103–114, 1996.

[34] M. Zaharia, M. Chowdhury, and J. M. Franklin, "cluster computing with working sets," *HotCloud*, vol. 10, 10 pages, 2010.

[35] M. Zaharia, M. Chowdhury, and T. Das, "Fast and interactive analytics over Hadoop data with Spark," *USENIX Login*, vol. 37, no. 4, pp. 45–51, 2012.

[36] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica, "GraphX: A resilient distributed graph system on spark," in *Proceedings of the 1st International Workshop on Graph Data Management Experiences and Systems (GRADES '13)*, June 2013.

[37] L. Gu and H. Li, "Memory or time: Performance evaluation for iterative operation on hadoop and spark," in *Proceedings of the Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, pp. 721–727, 2013.

[38] M. Zaharia, M. Chowdhury, and T. Das, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster

computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association*, p. 2, 2012.

[39] J. S. Greenfeld, "Matching GPS observations to locations on a digital map," *Transportation Research Board 81st Annual Meeting*, 2002.

[40] "General principles of civil law, [EB/OL]," http://www.law-lib.com/law/law_view.asp?id=221001.

[41] "Law of tort liability, [EB/OL]," http://www.npc.gov.cn/huiyi/cwh/1112/2009-12/26/content_1533221.htm.

[42] "Road Traffic Safety Law, [EB/OL]," http://www.npc.gov.cn/npc/xinwen/2011-04/23/content_1653570.htm.

[43] J. Hirten and S. Beroldo, "Ridesharing programs cost little, do a lot," *Transportation Quarterly*, vol. 51, no. 2, pp. 9–13, 1997.

[44] M. Naor, "On fairness in the carpool problem," *Journal of Algorithms. Cognition, Informatics and Logic*, vol. 55, no. 1, pp. 93–98, 2005.

[45] R. B. Noland, W. A. Cowart, and L. M. Fulton, "Travel demand policies for saving oil during a supply emergency," *Energy Policy*, vol. 34, no. 17, pp. 2994–3005, 2006.

[46] M. Ajtai, J. Aspnes, M. Naor, Y. Rabani, L. J. Schulman, and O. Waarts, "Fairness in scheduling," *Journal of Algorithms. Cognition, Informatics and Logic*, vol. 29, no. 2, pp. 306–357, 1998.