A Method for Real-time Trajectory Monitoring to Improve Taxi Service Using GPS Big Data

Zuojian Zhou^{a,b}, Wanchun Dou^{a,b,*}, Guochao Jia^{a,b}, Chunhua Hu^c, Xiaolong Xu^{a,b}, Xiaotong Wu^{a,b}, Jingui Pan^{a,b}

^aThe State Key Laboratory for Novel Software Technology, Nanjing University, China, 210023.

^bThe Department of Computer Science and Technology, Nanjing University, China, 210023.

^cThe School of Computer and Information Engineering, Hunan University of Commerce, Changsha 410205, China

---

Abstract

As taxi service is supervised by certain electronic equipment (e.g., GPS equipment) and network technique (e.g., cab reservation through Uber in USA or DIDI in China), taxi business is a typical electronic commerce mode. For a long time, taxi service is facing a typical challenge, i.e., passengers may be detours overcharged by some unethical taxi drivers, especially when traveling in unfamiliar cities. As a result, it is important to detect taxi drivers' misbehavior through taxi's GPS big data analysis in a real-time manner for enhancing the quality of taxi services. In view of this challenge, an online anomalous trajectory detection method, named OnATrade (pronounced 'on a trade', which means activities in a taxi trade on the fly), is investigated in this paper for improving taxi service using GPS big data. The method mainly consists of two steps: route recommendation and online detection. In the first step, route candidates are generated by using a route recommendation algorithm. In the second step, an online anomalous trajectory detection approach is presented to find taxis that have driving anomalies. Experiments evaluate the validity of our method on large-scale, real world taxi GPS trajectories. Finally, several value-added applications benefiting from big data analysis over taxi's GPS datasets are discussed for potential commercial applications.

Keywords: Taxi service, Trajectory detection, GPS big data, Behavior analysis

---

1. Introduction

In the past time, the decision-making in businesses is often challenged by incomplete, insufficient, and time-lapsed data. However, with the current exponential growth of electronic data available to businesses, more and more data sets often challenges enterprise or company's competition in data processing. Businesses now have more data than they can effectively use. For example, up until 2003, only 5 exabyte (1000 × 1000 × 1000 GB) of data were available, whereas nowadays the same amount of data can be created within two days in 2010 [1]. Currently, the capacity of big data analyzing is becoming a key basis of competition, which underpins new waves of productivity growth, innovation, and consumer surplus in business arena [2]. Since 2012, Big Data has become a research focus in academic, industry and government agencies for gaining competitive advantage. Typically, in January 2012, Big Data is a key theme of the World Economic Forum, and is highlighted in a report titled "Big Data, Big Impact: New Possibilities for International Development" [3].

As taxi service is supervised by certain electronic equipment (e.g., GPS equipment) and network technique (e.g., cab reservation through Uber in USA or DIDI in China), taxi business is a typical electronic commerce mode. According to statistics, in 2014, there are nearly 70,000 taxies running every day in Beijing, and about 55,000 in Shanghai. Therefore, Taxi services play a substantially important role in our daily life due to its door-to-door convenience. The GPS record of taxis make up of a big dataset. Mobile computing technology over GPS big data from GPS-equipped taxis make it possible to obtain

potential knowledge in understanding the behavior of urban commerce, the rule of social activities and road network dynamics [4, 5, 6, 7, 8, 9]. Moreover, various value-added applications, such as transportation management, city planning and personalized services [10, 11, 12, 13], could benefit from big data analysis over taxi's GPS datasets.

For a long time, taxi service is facing a typical challenge, i.e., passengers may be detours overcharged by some unethical taxi drivers, especially when traveling in unfamiliar cities. What's worse, a passenger may not be aware of a fraud when it is ongoing. As a result, it is important to detect taxi drivers' misbehavior through taxi's GPS big data analysis in a real-time manner for enhancing the quality of taxi services. For example, in San Francisco, a visitor who is not familiar with the city may want to travel from the University of California to the Asian Art Museum by taxi (see Fig. 1). The taxi driver knows the shortest route from the University of California to the Asian Art Museum, while the visitor does not know this information. In this situation, selecting a reasonable route depends on the taxi driver, and the passenger has no choice other than to stay in the taxi. Unfortunately, when a greedy taxi driver wants to commit taxi fraud during his service, the innocent passenger is not aware of this unreasonable behavior. Even if the visitor discovers the fraud later and then files a complaint, it is a difficult for the transportation bureau to obtain solid evidence for disclosing the fraud. Therefore, the transportation bureau faces additional management problems, especially in managing the taxi monitoring system. Technically, it is extremely difficult for the transportation bureau to obtain efficient fine-grain taxi supervision with the increasing number of taxis. A cunning and experienced taxi driver knows that it often costs the transportation bureau a substantial number of human resources to track complaints.

---

*Corresponding author

Email addresses: lygzjz@163.com (Zuojian Zhou), jackjianju@gmail.com (Guochao Jia), douwc@nju.edu.cn (Wanchun Dou), huchunhua777@163.com (Chunhua Hu), panjg@nju.edu.cn (Jingui Pan)

Figure 1: A Motivated Example

Preventing fraud before it is committed and finding evidence to prove fraud after it is committed are critical challenges, if we take into consideration the large number of taxis and the amount of GPS big data produced by the taxis.

With this observation, if there is a real-time and online line piloting service in the taxi, it could guarantee the QoS of the taxi service and the greedy taxi driver will be prevented from committing fraud. For example, if a taxi is equipped with a tablet that could display the reasonable routes in a visible way from the University of California to the Asian Art Museum, then when the driver turns in the wrong direction as indicated by the red arrow, the service system would warn the driver in time. At the same time, the warning information would be shared with the transportation bureau online. As a result, the online and real-time trajectory anomaly detection of taxis would play an important role in the social behavior analysis of a driver.

Traditional anomaly detection methods include the classical distance-based method [14, 15, 16, 17], density-based method [18, 19], distribution-based method [20], and deviation-based method [21]. These traditional methods are designed on the basis of spatial relational tuples to detect anomalies. In recent years, machine learning and data mining technology have been used to solve anomaly detection problems [22, 23, 24]. These methods mainly focus on discovering anomalies that have occurred. When considering the motivated example, we have determined that these methods are not suitable for online detection.

In view of these challenges, we propose an online anomalous trajectory detection method, named OnATrade (pronounced ‘on a trade’, which means activities in a taxi trade on the fly), in this paper for improving taxi service using GPS big data. The method mainly consists of two steps, *i.e.*, route recommendation and online detection. In the first step, a set of route candidates from a start point to a destination point (*i.e.*, the origin position and the destination position as demonstrated in Fig. 1) are discovered from a large number of historical trajectories. In the second step, online detection is conducted in real time by comparing the current ongoing trajectory with the route candidates. Additionally, the driving activity of taxi drivers is monitored in a real-time way. Once anomalous driving behavior is detected, feedbacks are released to the passenger, the driver and the transportation bureau. Finally, the abnormal driving behaviors would be prevented in real time.

The major contributions of our method are summarized as follows.

- We develop a real-time taxi trajectory monitoring method to detect online anomalous driving behavior online to prevent irregular or illegal driving behaviors in real time.
- The statistical data of the irregular or illegal driving behaviors are helpful for educating or training of the taxi drivers by the transportation bureau based on different profiles.

The remainder of this paper is organized as follows. Preliminary knowledge and problem definition are presented in Section 2. A real-time taxi trajectory monitoring method, named OnATrade, is investigated in Section 3. In Section 4, experiments are designed to evaluate the efficiency of our method. Section 5 discusses the extended commercial application. Related works and comparison analysis are presented in Section 6. Finally, the conclusion and our future work are presented in Section 7.

2. Preliminary Knowledge

2.1. Road Network Modeling

The large number of GPS sensors that are located on a taxi generate a substantial amount of GPS raw data. A record of GPS raw data contains a large amount of profile information for the current taxi state, such as taxi identification, longitude, latitude, speed, timestamp and the flag marking whether this taxi is occupied. Although GPS raw data have the location data of a taxi, improper working state of GPS devices often results in a deviation in taxi GPS locations. Moreover, abnormal GPS transmission may produce irregular taxi trajectories. Therefore, GPS raw data should be filtered when modeling the road network and the spatial environment where the taxi trajectory exists. For modeling the road network and the spatial environment, there are mainly two popular approaches, *i.e.*, grid decomposition and digital map modeling [25].

Grid decomposition is a naive and common method. Technically, in this method, equal-sized grids are generated to divide the city into small regions by using a user-defined parameter Θ . GPS points of trajectories could be mapped into these cell grids. After grid decomposition, trajectories of taxis could be described as a sequence of grids. In practice, some issues often influence its effective use. For example, if the grid size is not suitable, a sparse grid cannot model a taxi trajectory in a precise way. Digital map modeling could represent regular taxi moving patterns thanks to its spatial modeling method based on moving objects and basic road information. This modeling method describes the road network of the city in an efficient and explicit way.

In this paper, digital map modeling is introduced for road network modeling. Concretely, the road network modeling method is unfolded on an open source digital map, *i.e.*, OpenStreetMap (OSM). OSM is a collaborative project to create a free editable map of the world [26, 27]. OSM data are collected through manual surveys, GPS devices, aerial photography, and other free sources from large number of

Figure 2: Road Network Modeling

registered users. This crowd-sourced data are available under the Open Database License. In terms of data format, OSM uses a topological data structure, including four core elements: nodes, ways, relations and tags. **Nodes** are points that represent a geographic position and are stored as coordinates (pairs of a latitude and a longitude). **Ways** are a series of ordered nodes that represent a polyline, or possibly a polygon if they form a closed loop. Important usages of ways describe linear features such as roads and areas. **Relations** are ordered lists of nodes, ways and relations that are used for representing the relationship of existing nodes and ways. **Tags** are key-value pairs ($\langle \text{key}, \text{value} \rangle$). And metadata (such as their type, name and physical properties) regarding the map objects is stored through these tags. Based on this basic information of a target city, a road network could be modeled according to specific rules.

Given the OSM data of a target city, we could obtain all the core elements of the city. Let $\text{Node}_i \in \mathbb{N}$, $\text{Way}_i \in \mathbb{W}$ be the node and the way of the city ($1 \leq i \leq \text{num}_{\text{node}}$, $1 \leq j \leq \text{num}_{\text{way}}$), respectively. According to the tags of ways, roads could be filtered by specific key-value pairs, such as $\langle \text{highway}, \text{motorway} \rangle$, $\langle \text{highway}, \text{trunk} \rangle$, $\langle \text{highway}, \text{primary} \rangle$. There are two important concepts in the road network modeling process defined as follows.

Definition 1. A *segment* (seg) is a basic unit for each way generated by its consecutive and ordered nodes. The length of a seg is a short Euclidean distance denoted as $\|\text{seg}_{A,B}\| = \text{dist}(\text{Node}_A, \text{Node}_B)$ and its direction is from Node_A to Node_B .

$$\text{dist}(\text{Node}_A, \text{Node}_B) = \sqrt{(\text{lat}_A - \text{lat}_B)^2 + (\text{lng}_A - \text{lng}_B)^2},$$

where lat_A and lat_B denote the latitude of Node_A and Node_B , lng_A and lng_B denote the longitude of Node_A and Node_B .

Therefore, Way_j could be described as $\text{Way}_j = \{\text{seg}_{1,2}, \text{seg}_{2,3}, \dots, \text{seg}_{m-1,m}\}$ if Way_j has consecutive nodes, $\text{Node}_1 \text{Node}_2 \dots \text{Node}_m$. Then, the city has a segment set denoted as $\mathbb{G}$. Let $\mathbb{I}$ denote the set of nodes that are intersections of roads in the city. Due to the similar moving behavior of cars between two intersections, we enhance the

road network model by **section**.

Definition 2. A *section* (sec) is a series of segs (maybe one seg) generated by nodes that are intersections of roads, and its direction could be inferred from its segs ' direction. Given a sequential nodes $\text{Node}_1 \dots \text{Node}_i \dots \text{Node}_j \dots \text{Node}_m$ ($1 \leq i < j \leq m$), where $\text{Node}_i \in \mathbb{I}$, $\text{Node}_j \in \mathbb{I}$, then

$$\text{sec}_{i,j} = \{\text{seg}_{i,i+1}, \dots, \text{seg}_{j-1,j}\},$$

where we denote $\text{seg}_{k,k+1} \in \text{sec}_{i,j}$ ($i \leq k \leq j$). $\|\text{sec}_{i,j}\|$ stands for the distance of $\text{sec}_{i,j}$ and $\|\text{sec}_{i,j}\| = \sum_{m=i}^{j-1} \|\text{seg}_{m,m+1}\|$.

Let $\mathbb{C}$ be the set of all the sections of the city. Define function $\text{belongTo}: \mathbb{G} \rightarrow \mathbb{C}$ which could map known seg to a certain sec . For a better understanding, Fig. 2 presents the road network model of San Francisco in the area around $37^\circ 45' 59.76''N$ to $37^\circ 46' 28.2''N$ and $122^\circ 24' 35.64''W$ to $122^\circ 25' 1.2''W$. As shown in Fig. 2, red points stand for normal Nodes and blue points stand for the intersections of streets. A road may be divided into several parts by some intersections. For example, Way_1 is supposed to have consecutive nodes as $\text{Node}_A \text{Node}_B \dots \text{Node}_H$, where Node_A and Node_H are intersections. Thus, Way_1 could be represented as $\text{Way}_1 = \{\text{seg}_{A,B}, \text{seg}_{B,C}, \dots, \text{seg}_{G,H}\}$, and $\text{sec}_{A,H} = \{\text{seg}_{A,B}, \text{seg}_{B,C}, \dots, \text{seg}_{G,H}\}$ in which $\text{sec}_{A,H} = \text{belongTo}(\text{seg}_{E,F})$.

2.2 Trajectory Modeling

The sequential records of GPS raw data produced by a taxi's GPS device could be depicted as an important taxi operation status. A GPS trace of a taxi consists of a consecutive GPS points extracted from GPS raw data. Therefore, a taxi's moving trajectory could be generated by connecting all these GPS points. Since the driving anomaly frequently occurs in taxis that are occupied by passengers, we mainly focus on the taxi trajectories that are generated by occupied taxis in this paper. Formally, we define the concept of GPS points as follows.

Definition 3. A *GPS point* (p) is a triple denoted as $\langle \text{lat}, \text{lng}, \text{timestamp} \rangle$ which stands for the latitude, longitude and GPS generation time of p . Specially, the origin and destination of a taxi trajectory, i.e., the start place and the destination place, and denoted as $\text{Start} - P$ and $\text{Desti} - P$, respectively. A pair of $\text{Start} - P$ and $\text{Desti} - P$ would be abbreviated with SP and DP in this paper.

Let $\mathbb{P}$ be the set of all the GPS points generated by taxis in the city.

Definition 4. A *taxi trajectory* (tr) is an ordered series of GPS points which are generated by an occupied taxi from S to D .

$$\text{tr} = \{p_1 \rightarrow \dots \rightarrow p_i \rightarrow \dots \rightarrow p_n\}$$

where p_1 is S and p_n is D . And $p_i \in \text{tr}, p_i \in \mathbb{P}$ ($1 \leq i \leq n$).

Figure 3: Trajectory Modeling

Because the taxi is on the street or the road, every GPS point produced by taxis could be **assigned** to a certain section. Because the section maybe a polyline, the GPS point should be assigned to a segment to reduce the calculation complexity. As shown in Fig. 3, the black lines stand for the road, the blue points stand for nodes of intersections, the red point stands for normal node, and the green points stand for taxi GPS points with driving direction. The aim of assigning a GPS point is to find a segment which has the nearest distance with it. For the sake of simplicity, we show a simple example. As for a GPS point p_i in the assigning process, segments around p_i within a certain range are taken into consideration. The distances of p_i to $seg_{A,B}$ and to $seg_{B,C}$ are l and l' . As $l < l'$, we assign p_i to $seg_{A,B}$. In the same way, p_j is assigned to $seg_{B,C}$. We define function assign: $\mathbb{P} \rightarrow \mathbb{G}$ which stands for assigning a specific GPS point p to a segment seg in the city, which is denoted as $p \triangleright seg$. Therefore, $p_i \triangleright seg_{A,B}$ and $p_j \triangleright seg_{B,C}$. As a result of $seg_{A,B} \in sec_{A,C}$ and $seg_{B,C} \in sec_{A,C}$, both of p_i and p_j could be represented as $p_i \triangleright sec_{A,C}$ and $p_j \triangleright sec_{A,C}$.

After assigning GPS points to segments, we could transform the taxi trajectory to an **abstract trajectory**, which is easy for online anomaly score calculation in the following sections.

Definition 5. An **abstract trajectory** (atr) is a series of sections that are generated by the following process. Given $tr = \{p_1 \rightarrow \dots \rightarrow p_i \rightarrow \dots \rightarrow p_n\} (1 \leq i \leq n)$, a set of ordered sections could be produced by the assigning process of these GPS points.

$$atr = \{sec_1 \rightarrow \dots \rightarrow sec_i \rightarrow \dots \rightarrow sec_m\},$$

where $\omega = \{sec: sec = belongTo(assign(p_i)) \wedge p_i \in tr\}, sec_j \in \omega$ which could be expressed by $p_i \triangleright sec_j$ in an easy way ($1 \leq i \leq n, 1 \leq j \leq m$). In addition, the direction of each sec_j could be inferred through the moving direction of p_i in tr . The distance of atr is $|atr| = \sum_{i=1}^m dist(sec_i)$ and m could be represented as $|atr|$ to indicate the number of secs in atr .

In real life, the GPS points reported by taxis often have a low-sampling-rate problem [28]. This problem brings the uncertainty of the secs traversed by a taxi and possibly generates gaps among secs in atr . Therefore, an **augmenting** process is proposed to solve this problem. The augmenting process is motivated by the assumption in which taxi drivers go through those gaps with the shortest paths, so we augment atr

Figure 4: Complete Augmenting Process. (a) Raw GPS points of a tr . (b) Assigning process. (c) Augmenting process.

Figure 5: Overview of our method

with the secs that are traversed with shortest distance among gaps using the shortest path algorithm. As shown in Fig. 4(a), the yellow points indicate the GPS points of a taxi trip that starts from S and ends at D . In Fig. 4(b), the discontinuous blue lines stand for the secs which GPS points are assigned. In Fig. 4(c), a complete atr is generated in which the red lines represent the gap secs fixed through the augmenting process.

By assigning the GPS points to sections, we could transform the trajectory into section series instead of consecutive points. Henceforth, we will handle with both tr and atr in the rest of our paper.

3. A Real-time Taxi Trajectory Monitoring Method

With the above preliminary knowledge, an online anomalous trajectory detection algorithm, named OnATrade, is presented in this section. The application logic of our method is demonstrated in Fig. 5. Fig. 6 specifies the method for OnATrade in detail. As specified in Fig. 6, the method consists of two phases. The focus of the first phase is on data preparation offline. The result of data pre-processing is helpful for online anomalous trajectory detection, which is investigated in the second phase. The second phase aims to develop the algorithms related to route recommendation and online detection based on the previous modeled road network and trajectories. These algorithms put our OnATrade method into

Table 1: Notational and symbolic conventions

Symbol	Description
P_{cur}	Current GPS local of an ongoing taxi
tr_{cur}	Ongoing taxi trajectory
atr_{cur}	Abstract taxi trajectory generated by tr_{cur}
Rec	$atrs$ generated by route recommendation process
θ	Anomaly score of atr
φ	Enhanced anomaly score of atr

practice by providing real-time anomalous trajectory detection and online taxi anomalous behavior analysis. To facilitate further discussion, several related symbols are listed in Table 1.

3.1. Data Preparation

Given the large number of trajectory GPS points, a large number of valid taxi SD pairs could be obtained and could represent popular places in people’s daily life. Therefore, these historical trajectories could provide a strong evidence for predicting taxi routes that have the same SD information. Before the online phase, we should generate GPS grid distribution for our target city. We split the city by deploying grid decomposition. Then, GPS points are scattered in those equal-sized grid-cells that have unique identifier gid . This scattering process could be described by function $mapping(p)$ where p indicates a GPS point and returns its corresponding gid . We only focus on those SD points that directly indicate the pick-ups and drop-offs. As depicted in Fig. 7 (Indexing Table Generation), we allocate every historical atr with gid and mark its SD information. Therefore, we obtain a structured mapped indexing table T_{index} that contains three columns containing grid-cell gid , the atr identifier and the SD type. Hence, we define function $query(gid_o, gid_d)$ which could easily obtain a collection of $atrs$ from T_{index} when certain SD grid-cell information is given.

3.2. Route Recommendation

A baseline for route recommendation should be a shortest and feasible path. In practice, there may be more than one shortest path and a feasible path. For two paths that have a same length, they may cover different routes in practice. For example, for a rectangle, there are two paths between two vertexes connected by a diagonal. The two paths cover different sides of the rectangle, even though the two paths have the same length. In a city’s road network, there are many similar situations for a trip from one place to another. As a result, for as SD pair, there may be two or more paths that have nearly same length but cover different routes caused by taxi drivers’ different preferences. In our method, the path selected by most of the taxi derives in the past time is treated as a reasonable one in our method. The top 1 path recommended by our method could be treated as an optimal path.

To recommend some feasible path for taxi drivers, popular route patterns could be learned from historical trajectories in the online phase. In our method, for a coming trip between two places, a history taxi trajectory, which has the same start place and destination place with the coming trip, could be recruited

Phase 1: Data preparation (offline). Data preparation aims to provide a detailed easy-to-use infrastructure for the online process. In this phase, Road network modeling and trajectory modeling are conducted based on OSM map data and raw GPS data. Combined with these data, GPS grid distribution is generated to indicate popular regions where people often depart.

Phase 2: Online Anomalous Trajectory Detection (online). Based on the previous models, an online anomalous trajectory detection algorithm, named OnATrade, is developed and consists of two steps.

Step1: Route Recommendation. According to the specific S and D of a taxi trip, several reasonable and suitable routes could be generated based on historical trajectories. The taxi driver chooses one of them to direct his trip.

Step2: Online Detection. Ongoing taxi trajectory is processed in real time, and online behavior analysis of this taxi ride could be provided to indicate whether this ongoing trajectory is anomalous. Furthermore, passengers or the transportation bureau could participate in this process to improve the quality of public service. Passengers could give rating or feedbacks with a taxi-equipped tablet, and transportation bureau could have could intervene if emergency occurs.

Figure 6: Specifications of our Method

as a referred trip for route recommendation. A path associated with a definite trip between two places is treated as reasonable if it is selected by most of the taxi derives in past time. Here, we will investigate the first step, *i.e.*, online route recommendation of our method OnATrade.

3.2.1. Filtering Trajectories Process

When a passenger enters a taxi, S is automatically located at the current position of a taxi trip and D is confirmed through an intelligent taxi-equipped tablet by a passenger. After current SD information is obtained, these two positions are mapped into grid-cells instantly by function $mapping$ shown in line 2 in Algorithm 1. The process of filtering trajectories is performed by retrieving T_{index} efficiently through function $query$ with parameters obtained from the previous step. The filtered atr collection is the basis of route recommendation and is given as $S_{filter} = \{atr_1, \dots, atr_n\}$.

3.2.2. Popular Route Generation

Filtered trajectories could give valid evidence on the driving behavior of taxi drivers between pre-confirmed SD information. Hence, the task of route recommendation is to find those routes that most drivers prefer to go through with given SD pairs. Let Rec be the set of recommended routes and let k denote the number of recommended routes. Given S and D , the output of this process is $Rec = \{atr_1, \dots, atr_k\}$

Figure 7: Application Framework of Our Method

in which $atr_i (1 \leq i \leq k)$ is a complete abstract trajectory from S to D and guides driving activity. Because abnormal taxi trajectories are always few and different from other trajectories, they are simple and easy to implement to find many similar trajectories. This constitutes the basis of popular route generation.

Here, we denote each abstract trajectory atr as a word and the section sec in atr as a letter. In the process of popular route generation, we compute trajectory similarity by applying the Longest Common Subsequence (LCS) algorithm [29]. The LCS algorithm could find the longest common sec sequence of two compared atr s. For example, if $atr_1 = \{sec_1 \rightarrow sec_2 \rightarrow sec_3 \rightarrow sec_5 \rightarrow sec_7 \rightarrow sec_8 \rightarrow sec_9 \rightarrow sec_{10}\}$, $atr_2 = \{sec_1 \rightarrow sec_2 \rightarrow sec_3 \rightarrow sec_4 \rightarrow sec_5 \rightarrow sec_7 \rightarrow sec_9\}$, the longest common sec sequence of atr_1 and atr_2 is $sec_1 \rightarrow sec_2 \rightarrow sec_3 \rightarrow sec_5 \rightarrow sec_7 \rightarrow sec_9$. Therefore, $LCS(atr_1, atr_2) = 6$. We define the similarity of two atr s according to Equation 1. Naturally, $sim(atr_1, atr_2) = 6/7$.

$$sim(atr_a, atr_b) = \frac{LCS(atr_a, atr_b)}{\min(|atr_a|, |atr_b|)}, \quad (1)$$

The main steps of popular route generation are shown from line 5 to 21 in Algorithm 1. After S_{filter} is obtained, the size of S_{filter} has an important effect on the quality of route recommendation. When the number of historical trajectories

reaches a certain value, a reasonable and practical route recommendation will take effect. In our method, we set trajectory number threshold λ_{num} . If the number of S_{filter} is greater than λ_{num} , popular route generation is performed to produce recommended routes. Otherwise, route recommendation is done by deploying k -shortest path algorithm (KSP), a classic graph algorithm in computer science. In this paper, we use KSP algorithms proposed in [30, 31] and these algorithms have been successfully applied in many applications. However, if the distances of the recommended routes in Rec are greater than our predefined threshold λ_{dist} , reasonable and shorter routes will be filtered by this distance threshold (line 23, 24 in Algorithm 1)

The principal idea for popular route generation is to cluster similar atr s in S_{filter} and select the top k common routes from these clusters. For every atr in S_{filter} , it is easy to gain the most similar abstract trajectory in the set of candidate routes S_{cdd} through predefined similarity threshold as shown in line 9 in Algorithm 1. Nevertheless, if we do not find a similar atr in S_{cdd} , we add this atr to S_{cdd} as a new candidate route. After the clustering process, we sort these clusters in S_{cdd} in a descend order and choose the top k to generate recommended routes.

It is intuitive to deploy the KSP algorithm because taxi drivers usually choose the shortest distance path or the smallest time cost path to obtain maximum interest. We do not consider special cases, such as the passenger wants to pick up their friends or the passenger's other requirements. Therefore, when there are not plenty of historical trajectories between certain SD , we choose the KSP algorithm to produce recommended routes.

Based on the preliminaries mentioned in the previous sections, we could easily construct a digraph $\langle I, C \rangle$ where I and C are the set of intersections and sections of the city, respectively. There are mainly three steps in our recommendation process. Firstly, according to pre-confirmed OD and current taxi driving direction, $Node_O$ and $Node_D$ ($Node_O, Node_D \in I$) which are intersections are filtered by the nearest Euclidean distance with O and D . KSP actually operates with $Node_O$ and $Node_D$ to generate k shortest paths. Secondly, since every sec has its own highway class and different highway classes have various travel cost, we evaluate each sec through the aspect of distance cost and time cost. In distance cost evaluation, sec is weighted by its highway class denoted as w_{sec} and the cost of every recommended $atr_i (1 \leq i \leq k)$ is calculated according to Equation 2:

$$DisCost_{atr_i} = \sum_{sec \in atr_i} w_{sec} ||sec|| \quad (2)$$

where $||sec||$ stands for the length of sec in $atr_i (1 \leq i \leq k)$. And in time cost evaluation, the cost of each sec is estimated by the average speed denoted as v_{sec} which is collected by transportation bureau. Thus, in our KSP algorithm, the cost of every $atr_i (1 \leq i \leq k)$ is calculated based on Equation 3:

Algorithm 1 Route Recommendation

Input: T_{index} - the structured mapped indexing table; SD - origin and destination of current taxi trip; λ_{num} , λ_{dis} , λ_{sim} - number filter threshold, distance threshold, and the similarity threshold among $attrs$.

Output: Rec - the set of k recommended routes.

- 1: $S_{cdd} \rightarrow \emptyset$; // S_{cdd} is the set of candidate $attrs$.
/*Filtering Trajectories*/
- 2: $gid_S = mapping(S)$, $gid_D = mapping(D)$;
- 3: $S_{filter} \leftarrow query(gid_S, gid_D)$;
- 4: $n = sizeof(S_{filter})$; // n is the number of $attrs$ in S_{filter} .
/*Popular Route Generation*/
- 5: **if** $n > \lambda_{num}$ **then**
- 6:   Choose one atr from S_{filter} randomly and add it to S_{cdd} ; // S_{cdd} initialization
- 7:   **repeat**
- 8:   Select one atr_i from S_{filter} ;
- 9: $\lambda = \max(sim(atr_i, atr_j))$, $atr_j \in S_{cdd}$;
- 10:   **if** $\lambda < \lambda_{sim}$ **then**
- 11: $S_{cdd} \leftarrow S_{cdd} \cup atr_i$;
- 12: $num_{atr_i} = 1$; // num_{atr_i} is the number of atr_i added in S_{cdd} .
- 13:   **else**
- 14: $num_{atr_i} = num_{atr_i} + 1$;
- 15:   **end if**
- 16:   remove atr_i from S_{filter} ;
- 17:   **until** S_{filter} is empty;
- 18:   Sort S_{cdd} by num_{atr} in descend order;
- 19: $Rec \leftarrow$ select top k $attrs$ in S_{cdd} ;
- 20: **else**
- 21: $Rec \leftarrow k$ Shortest Path Algorithm;
- 22: **end if**
- 23: $dis_{min} = \min(|atr_i|)$, $atr_i \in Rec$;
- 24: $Rec = \{atr: atr \in Rec \wedge |atr| \leq dis_{min}(1 + \lambda_{dis})\}$;

$$TimeCost_{atr_i} = \sum_{sec \in atr_i} \frac{|sec|}{v_{sec}} \quad (3)$$

where $|sec|$ stands for the length of sec in atr_i ($1 \leq i \leq k$). The final operations are to rank atr_i s by $DisCost_{atr_i}$ or $TimeCost_{atr_i}$ and to generate Rec .

Although route recommendation could generate k routes according to pre-confirmed SD pair, the performance of this route recommendation method might be improved because more external variables could be considered in addition to the distance cost and the time cost. For example, the real-time road condition is important for a taxi driver's driving activity because it affects the dynamics of the entire road network. Every taxi driver has his own way of taking passengers to the destination on an appropriate route. Specially, the passenger may have some natural requirements including picking up friends at a specific location or just following some car. Hence, enhanced route recommendation improvements could be developed based on the basic route recommendation algorithm. Three different aspects could be added to the enhanced route

recommendation, including real-time road condition estimation, a personalized behavior analysis of taxi drivers, and a customized passenger's requirements. Nevertheless, the focus of this paper is on the online detection process which is described in the following subsection. The enhanced route recommendation method will be discussed in our future research work.

3.3. Online Detection

A route collection Rec for taxi drivers is generated through the route recommendation process according to the given SD pair. Therefore, it is easy to perform online detection on the basis of these recommended routes.

3.3.1. Basic Online Detection

In the process of online detection, the routes in Rec are a guideline for the following driving activity. The detailed process of online detection is presented in Algorithm 2. From a fine-grained perspective, the taxi's current moving direction should be in accordance with the direction of most of $attrs$ in Rec . To accomplish the online anomalous trajectory detection, we first transform the current driving location p_{cur} to sec_{cur} due to $p_{cur} \triangleright sec_{cur}$ (line 4). Then, atr_{cur} is constructed by sec_{cur} to be a complete section sequence by using the augmenting process (line 5, 6). We maintain an anomaly score to evaluate the abnormal degree of the ongoing taxi trajectory on the basis of Equation 1. Presently, we could compute the ongoing taxi's anomaly score θ as shown in line 7.

Although the basic online detection method has high accuracy, it is sensitive to data noise. To improve the robustness of this basic method, an enhanced anomaly score ϕ , which evolves with travel time, is calculated in real time. Meanwhile, this score indicates the trend of the driver's overall driving activity during a taxi trip.

Algorithm 2 Online Detection

Input: Rec - the set of k recommended routes; tr_{cur} - ongoing trajectory which p_{cur} is the current incoming GPS point; $\theta_{anomaly}$ - trajectory anomalous threshold.

Output: ϕ - enhanced anomaly score; ψ -collection of anomalous GPS points

/*initialization*/

- 1: $\psi \rightarrow \emptyset$;
- 2: $atr_{cur} \leftarrow \emptyset$;
// atr_{cur} is abstract trajectory generated by tr_{cur} .
/*processing*/
- 3: **while** tr_{cur} is not complete **do**
- 4: $sec_{cur} = belongTo(assign(p_{cur}))$;
- 5: $atr_{cur} = atr_{cur} \cup sec_{cur}$; //update atr_{cur}
- 6:   augmenting atr_{cur} ;
- 7: $\theta = 1 - \max\left(\frac{LCS(atr_{cur}, atr_i)}{|atr_{cur}|}\right)$, $atr_i \in Rec$;
- 8:   **if** $\theta > \theta_{anomaly}$ **then**
- 9: $\psi \leftarrow \psi \cup p_{cur}$;
- 10:   **end if**
- 11: **end while**

Table 2: Experiment Context

	Client	HANA Cluster
Hardware	Lenovo ThinkpadT430 machine with Intel i5-3210M 2.50GHz processor, 4GB RAM and 250GB Hard Disk.	Master (1 node): HP Z800 Workstation Intel(R) Multi-Core X5690 Xeon(R), 3.47GHz/12M. Cache, 6cores, 2 CPUs, 128GB (8x8GB+ 4X16GB) DDR3 1066MHz ECC Reg RAM, 2TB 7.2K RPM SATA Hard Drive. Slave (1 node): HP Z800 Workstation Intel(R) Multi-Core X5690 Xeon(R), 3.47GHz/12M. Cache, 6cores, 2 CPUs, 128GB (8x8GB+ 4X16GB) DDR3 1066MHz ECC Reg RAM, 2TB 7.2K RPM SATA Hard Drive.
Software	Windows 7 Professional 64bit OS and HANA Studio.	SUSE Enterprise Linux Server 11 SP3 and SAP HANA Platform SP07.

3.3.2. Enhanced Anomaly Score

As the ongoing trajectory is moving over time, the historical moving behavior may influence the accuracy of current anomaly detection. Therefore, we enhance our anomaly score by adding the current taxi trip's historical anomaly scores in an evolving way. According to the anomaly score θ , we could judge whether the ongoing taxi trajectory is anomalous in a direct way. From the point of view of the entire taxi trip, we maintain an enhanced anomaly score φ combined with a historical score. This enhanced anomaly score is used to provide an efficient and accurate online decision-making basis for the transportation bureau and to rank anomalous trajectories once they are finished in a more comprehensive perspective. As the ongoing trajectory moves over time, historical anomaly scores have less influence than the current anomaly score. Hence, we have a weight coefficient τ to balance historical and current values. Suppose the time gap between any two neighboring GPS entries (the timestamp when taxi GPS points are received) is equal to Δt , and tr_{cur} starts at initial time t_0 , the anomaly score of tr_{cur} at t_0 is $\varphi_{t_0} = \theta_{t_0}$, and the anomaly score at time t_1 is $\varphi_{t_1} = (1 - \tau)\theta_{t_1} + \tau\varphi_{t_0}$. Therefore, at any time t_k , the anomaly score of tr_{cur} could be given as:

$$\varphi_{t_k} = (1 - \tau)\theta_{t_k} + \tau\varphi_{t_{k-1}}, \quad (4)$$

In another way, Equation 4 could be expanded as:

$$\varphi_{t_k} = \tau^0(1 - \tau)\theta_{t_k} + \tau^1\varphi_{t_{k-1}} + \dots + \tau^{k-1}(1 - \tau)\theta_{t_k} + \tau^k\varphi_{t_0}. \quad (5)$$

As a result, Equation 5 could be presented simply as Equation 6:

$$\varphi_{t_k} = (1 - \tau) \sum_{i=0}^k \tau^{k-i} \theta_{t_i} + \tau^{k+1} \theta_{t_0}. \quad (6)$$

Intuitively, φ is higher when the ongoing taxi trajectory is away from $atrs$ in Rec . Therefore, the enhanced anomaly score φ is an efficient indicator for the online detection process, and detailed analysis is conducted in a fine-grained view. Through online anomaly scores, the transportation bureau could easily analyze the motion pattern of an online

(a) Origin distribution

(b) Destination distribution

Figure 8: Quantitative Distribution of Taxis SD Pairs in San Francisco

Figure 9: Route Coverage Ratio of Route Recommendation

anomalous taxi. Moreover, an additional social behavior analysis of online taxis is generated through online data to improve the efficiency of management for taxi companies.

4. Experiment and Evaluation on In-Memory Database

4.1. Experiment Data Sets and Experiment Context

To evaluate the efficiency and effectiveness of OnATrade, both real-world data and synthetic data are adopted in the

Figure 10: Visualization of Taxi Trajectories Based on SD_1 Pair and its Route Recommendation Result with Different k values equal three, five, and seven (left to right).}

experiments. This real-world data set [32] contains more than 10 million records of taxi cabs' mobility traces in San Francisco, USA, which is provided by Michal Piorkowski, Natasa Sarafijanovic-Djukic and Matthias Grossglauser. It contains GPS coordinates from approximately 500 taxis that was collected over 30 days from May to Jun of 2008 in the San Francisco Bay Area. Based on the moving behavior of taxi drivers, synthetic data are added to enhance our online analysis of our method.

Technically, our experiments are conducted in a HANA cluster environment. The services recruited in our experiment are distributed in the cluster. Because we make full use of in-memory database, our method can be processed efficiently in a short response time. Specific hardware and software configurations are listed in Table 2.

4.2. Experiment Data Sets and Experiment Context

Two groups of experiments are conducted to measure the rationality of route recommendation and the efficiency of online detection for OnATrade. In the first group, we start by analyzing the quantitative distribution of the origin and destination of taxi trajectories in real-word data set. Then, the route coverage ratio and the time consuming route recommendation are presented in a detailed description. In the second group, we explore the efficiency and additional analysis of online detection for OnATrade with ample historical and synthetic taxi trajectories. In the experiment, we set up route recommendation with k equal to three, five, and seven, respectively. The w_{sec} , v_{sec} of each sec is an empirical value according to its way type. The number filter threshold λ_{num} , the distance threshold λ_{dis} , and the similarity threshold λ_{sim} are equal to 500, 0.6173, 0.8742, respectively, and based on the experiment context. Moreover, the weight coefficient τ is set to be 0.43729 according to experimental analysis.

Route Recommendation Analysis. For a better experiment description, we first build heatmaps to present the quantitative distribution of taxi SD pairs. Then, hot SD pairs are selected as the origins and the destinations of typical GPS traces in our experiment. As the heatmaps are shown in Fig. 8, most of the passengers enter and exist taxis in the main urban area. Therefore, five representative SD pairs are chosen as the experimental origins and destinations, and the number of

Table 3: Representative SD -pairs Information

	#trajectories	S GPS location	D GPS location
SD_1	1505	37.78605°N, 122.41104°W	37.80599°N, 122.41859°W
SD_2	845	37.76077°N, 122.43507°W	37.78734°N, 122.41327°W
SD_3	728	37.79324°N, 122.39769°W	37.77967°N, 122.41435°W
SD_4	1007	37.80287°N, 122.43773°W	37.79551°N, 122.39958°W
SD_5	895	37.80034°N, 122.43956°W	37.78927°N, 122.41555°W

historical trajectories filtered by these SD pairs is listed in Table. 3. Moreover, Fig. 10 shows the visualization of the trajectories based on SD_1 and its corresponding route recommendation results when k equals three, five, seven, respectively. In Fig. 10, the green point and the red point denote the center of S and D , respectively. The blue lines with the arrow represent the driving direction, and historical taxi trajectories' SD pairs are in the circle whose center is SD_n ($1 \leq n \leq 5$) and whose radius is 500 meters. Moreover, the routes in the recommendation result may have overlapped sec s. Based on these ample trajectories and route recommendation results, the route coverage ratio is calculated to verify the rationality of the route recommendation results according to the following equation.

$$r_c = \frac{|C_{rec}|}{|C_{cdd}|}, \quad (7)$$

where $C_{cdd} = \{sec \mid sec \in atr \wedge atr \in S_{cdd}\}$ and $C_{rec} = \{sec \mid sec \in atr \wedge atr \in Rec\}$. S_{cdd} is the set of candidate $atrs$ generated by the popular route generation process in route recommendation (Algorithm 1) with certain SD pair, and Rec is generated based on S_{cdd} with a specific k value. $|C_{rec}|$ and $|C_{cdd}|$ are the number of sec s in C_{rec} and C_{cdd} , respectively.

As depicted in Fig. 9, the route coverage ratio of route recommendation increases as k increases. With the consideration of time consuming in the route recommendation process shown in Fig. 11, the higher k is, the greater the time cost of route recommendations is. Therefore, combined with the above aspects, we could obtain a rational route recommendation result with $5 \leq k \leq 7$ in a better performance.

Figure 11: Time Consuming of Route Recommendation

Figure 12: Running Example of Online Detection with SD_1 Pair and its Route Recommendation Result with k equals five

Online Detection Analysis. In this part, ample historical and synthetic taxi trajectories have been tested with the online detection process. As an example, we select SD_1 to show the detailed detection process. As depicted in Fig. 12, the green and red points are S and D , respectively. The blue lines indicate all of the routes in Rec based on the predefined SD with $k = 5$. The red line is the actual ongoing trajectory atr_{cur} and the taxi's GPS points are yellow. Moreover, the red lines with an arrow show the ongoing taxi driving direction. This case is conducted without the feedback of passengers temporarily. The threshold of enhanced anomaly score $\varphi_{anomaly}$ is set to 0.13127 which is an experimental value according to large amount of online detection trials.

As shown in Fig. 13, the first five GPS points are obviously normal. As the taxi driver drives along one of $attrs$ in Rec , both θ and φ are zero, which indicates that the driver's moving behavior is normal. However, at the sixth GPS entry shown in Fig. 12, the taxi driver is deviating from all of the $attrs$ in Rec . As a consequence, both θ and φ increase and θ exceeds φ while φ only has a comparable and slighter raise.

As the taxi continues moving over time, θ and φ become larger and φ always changes more mildly than θ , which

Figure 13: Anomaly Scores in Online Detection Process with SD_1 Pair and its Parameter k equals five

Figure 14: The ROC curves of OnATrade

implies that the enhanced anomaly score φ is more robust than anomaly score θ . Until φ approximates to threshold $\varphi_{anomaly}$, this ongoing trajectory will be reported as anomalous and this anomaly will also draw the attention of transportation bureau for further observation. In Fig. 13, from the eighth GPS entry of an ongoing trajectory, φ is always above $\varphi_{anomaly}$ which means that the on-going trajectory is anomalous from the eighth GPS point. The red dashed line in Fig. 12 indicates that the anomalous ongoing trajectory is under the close watch of transportation bureau and its taxi company since it reaches the online detection anomaly threshold. As a result, this taxi driver will be under the full control of relevant departments. Suppose there is no feed-back from passenger. Thus, this trajectory will be reported to the transportation bureau and enhanced anomaly scores will continue to be generated to provide evidence of moving behavior. Moreover, at the 15th GPS entry in Fig. 12, this ongoing trajectory is back to one atr in Rec while θ and φ decrease at the same time. The decreasing trend of θ and φ implies that the taxi driver drives back to normal routes.

Based on the large number of experimental trials, we could find that it is more inconspicuous for long routes to detect taxi frauds timely but that short route is sensitive for detouring. Taking an insight into the definition of θ and φ , it is easy to notice that the LCS of an ongoing trajectory for a long route

has less effect and is less sensitive due to its long distance.

Combined with passengers' feedback, Fig. 14 shows the ROC curves of each *SD* dataset. A ROC curve is an integrated indicator that reflects the continuous variables of a true positive rate (detect anomalous trajectories successfully) and a false positive rate (normal trajectory detect as anomalous). The greater the area under the curve is, the higher the detection accuracy is. On the ROC curve, the coordinates of the point closest to the up-per left of figure have a higher threshold for the true positive rate and the lower false positive rate. For all datasets, it is simple to find that OnATrade can achieve a high detection rate.

As all mentioned above, our method could efficiently detect anomalous trajectories while the trajectories are ongoing, and the feedbacks from passengers could be easily confirmed to improve the interaction of entire taxi trip.

5. Application Analysis

With the data persistence of online detection data, we obtain a more comprehensive understanding of taxis' society behavior. This analysis has wide commercial application value as discussed in the following profiles.

5.1. Case 1: Real-time Public Traffic Supervision over Taxis

With the real-time supervision of taxis running all over a city, the transportation bureau could obtain efficient management over a large number of taxis and could protect passengers from taxi frauds. For a driver, if he or she has a green hand, the method presented in this paper is a real-time route pilot in practice. With the help of the real-time route pilot, his or her driving ability could improve over time. If the driver is apt to trying fraud, the method presented in this paper could stop the fraud in time before the passenger is hurt. The times for a driver's abnormal behavior show the tendency for fraud or driving ability, which directly disclose the QoS of a taxi. If the times of a driver's abnormal behavior reach a certain number, warnings or persuasions can be delivered to him or her so that his or her service can be improved.

5.2. Case 2: QoS Improvement of Taxi Service

In taxi companies, the analysis of taxi driver behavior based on our online detection method could provide an efficient evaluation of the driver's performance. The officers in taxi companies could perform reasonable rating activities to achieve effective management. Moreover, our online detection method could help taxi companies build a good service environment with fine, self-disciplined taxi drivers. Additionally, it is helpful for taxi companies to obtain better management from a fine-grained perspective. In traditional management, there are a few efficient methods that deploy the measures mentioned above. If a passenger complains about taxi fraud, he has no choice but to call the taxi company. However, the process for evidence collection may be long. Compared to the previous traditional management, our online and real-time detection method could improve the QoS of taxi companies.

5.3. Case 3: Embed Advertisement Accompanied with Route Navigation

The method presented in this paper has wide commercial value for advertisements. The key issue in advertisements is exposing its content to its target consumer for more exposure and more value. For example, route recommendation could be sponsored by several advertisers in an embedded way, *e.g.*, embedded audio, embedded video or embedded text. Furthermore, the method presented in this paper could provide a novel way for advertisement release. Concretely, the passenger is a potential consumer of the advertisements along the candidate routes, in practice. If there is more than one candidate route for a passenger reach to his or her destination, it will cause different advertisers to sponsor different paths for their advertisements.

6. Related Work and Comparison Analysis

In this section, related works are briefly reviewed in mainly two relevant aspects: trajectory pattern analysis and anomalous trajectory detection. For trajectory pattern analysis, related research works focus on mining GPS data for various types of novel applications. Liu *et al.* [8] revealed cabdrivers' operation patterns by analyzing their continuous digital traces and they categorized taxi drivers by their daily income, which demonstrates the great potential to utilize the massive pervasive data sets and to finally understand human behavior and high-level intelligence. Yuan *et al.* [4] provided users with the fastest route by mining smart driving directions from a large number of historical taxi GPS trajectories. Wang *et al.* [9] proposed a framework of using a nonparametric Bayesian model for unsupervised trajectory analysis and semantic region modeling in surveillance settings by treating trajectories as documents and positions as words. In [5], Ziebart *et al.* presented *PROCAB* based on a taxi driver's driving experience in finding the best routes to a destination and in guiding users with driving directions by utilizing taxi GPS trajectories. In [6], trajectory patterns were proposed as concise descriptions of frequent behaviors in terms of both space and time. Zheng *et al.* aimed to mine interesting locations and classical travel sequences in a given geospatial region based on different users' GPS trajectories.

For anomalous trajectory detection, the related works mainly focus on anomalous trajectory detection which has a high correlation with our work. Bu *et al.* [33] proposed a framework for monitoring anomalies over continuous trajectory streams. Local clusters were built upon trajectory streams and efficient pruning strategies were used to detect anomalies. In [34], an evolving trajectory outlier detection method was proposed based on an evolving outlying score with continuous computation in view of evolving moving direction and the density of trajectories. Therefore, this method, which takes advantage of a decay function, could identify evolving outliers at a very early stage. In [35], a partition-and-detect framework for trajectory outlier detection was introduced that combines both distance and density information. Ge *et al.* [12] developed a taxi driving fraud detection system that integrates travel route evidence and driving distance evidence based on

Table 4: Comparison result in time consumption

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
OnATrade (Unit: ms)	17.86	19.98	18.71	19.66	18.50	19.87	19.24	19.03	19.45	18.92
iBoat (Unit: ms)	11105.9	11079.1	11074.7	11075.3	11099.9	11124.5	11081.2	11084.1	11066.4	11089.0

Dempster-Shafer theory. Similarly, in [10], an isolation-based anomalous trajectory method was presented, which mainly exploits intrinsic properties of anomalous trajectories: few in number and different from the majority. Moreover, taking advantage of the method presented in [10], Chen *et al.* [11] extended their research work to online anomalous trajectory detection. Li *et al.* [36] developed a temporal outlier detection method that aims to detect anomalies in vehicle traffic data through the aspect of road network traffic changes. Furthermore, the learning-based approaches presented in [22, 24, 23] are successfully applied in anomalous trajectory detection. In addition, more outlier detection methods are general and are not specially proposed for trajectory data. They are in different problem scenarios with specific anomaly detection method. They range from distance-based method [14, 15, 16, 17], density-based method [18, 19], distribution-based method [20] to deviation-based method [21].

The methods presented in these related works pay little attention to real-time service. For example, in [35], the classic outlier detection method has a good performance in an offline situation. However, for online taxi analysis, massive calculation prevents it from detecting outlier in real-time and in a highly efficient way. In [10], a taxi fraud detection system is investigated and is used to detect anomalous trajectories for offline taxi analysis. The work presented in [10] is also used in this offline scenario. In [11], an online anomalous trajectory detection method, named iBoat, is presented. In this method, passengers have a few interactions during detection process. Besides, the method presented in [22, 24, 23] needs data training which is expensive to label and has a high time cost in data preparation. In this situation, it is hard to complete the online and real-time abnormal trajectory monitoring. Some distance-based method [14, 15, 16, 17], density-based method [18, 19], distribution-based method [20] and deviation-based method [21] also face the same real-time and online detection problems mentioned above.

Compared with these methods mentioned above, OnATrade paper is well suited for online monitoring and could provide feedback in real time by focusing on online taxi analysis, which is helpful for improving the QoS of a taxi service in a novel way. Without loss of generality, a typical comparison analysis is investigated, here, between our method and the method iBoat presented in [11]. As demonstrated in Fig. 15, along a path indicated by a pair of $S_i D_i$, there are ten point, our method and the method iBoat will be used to determine if they are covered by the path or not. Time consumption is taken as the evaluation criterion. As a result, both of the two methods could achieve the goal. Table 4 indicates the time consumption point by point.

Figure 15: A path indicated by a pair $S_i D_i$

From Table 4, we could find that our method runs fast than iBoat does. Actually, as indicated in [11], the data processing for anomalous detection in their method is running in an offline way for shorting the response time. Our method is running in a really online way for satisfying the real-time requirement.

In theory, the time complexity of route recommendation, *i.e.*, Algorithm 1, and online detection, *i.e.*, Algorithm 2, are discussed in Page 7 in detail. Specifically, the time complexity of route recommendation in our method, *i.e.*, Algorithm 1, is $O(n)$, suppose that there are n history taxi trajectories associated with a concrete trip; the time complexity of online detection of our method, *i.e.*, Algorithm 2, is a $O(m)$ suppose that there are m GPS records associated with a concrete trip. For the classic graph algorithm, *i.e.*, k -shortest path algorithm (KSP), as discussed in [30, 31], its time complexity is $O(e + v * \log v + k * v)$, suppose that there are e edges and v vertex in a city's road network. As a result, our method is suitable for online and real-time applications. Besides, as the data processing in our method is enabled by a cloud platform, our system is scalable in practice.

7. Conclusions and Future Work

As taxi service is supervised by certain electronic equipment (e.g., GPS equipment) and network technique (e.g., cab reservation through Uber in USA or DIDI in China), taxi business is a typical electronic commerce mode. Mobile computing technology over GPS big data from GPS-equipped taxis makes it possible to obtain potential knowledge in understanding the behavior of urban commerce, the rule of social activities and road network dynamics. In this paper, we have proposed a real-time taxi trajectory monitoring method to detect taxi anomalous driving activities online and in real time. Technically, road network modeling is investigated first. Based on the road networking model, an online anomalous trajectory detection method, named OnATrade, has been presented to analyze the online driving behaviors of taxi drivers. This method is validated based on a large data set for real-world GPS traces. In the future, the method could be perfected for demonstrating its advantage in social behavior analysis. Moreover, more real-world applications will be developed to validate our method, such as mobile APP supporting smart travel, real-time path recommendation and navigation service in smart city development, *etc.* We believe that these value-added applications could benefit from our big data analysis method over taxi's GPS datasets.

Acknowledgement

This paper is partially supported by the National Science Foundation of China under Grant No. 91318301 and No.61273232, the Key Research and Development Project of Jiangsu Province under Grant No. BE2015154, the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University, the Program for New Century Excellent Talents in University under Grant NCET-13-0785, the project for Research on Real-Time Processing and Intelligent Analysis Technology of Electric Power Big Data from State Grid Corporation of China (SGCC), and in part by the Program for Hunan Provincial Key Laboratory of Mobile Business Intelligence.

Reference

- [1] X. Yi, F. Liu, H. Jin, Building a network highway for big data: architecture and challenges, *IEEE Network*, 28(4) (2014) 5-13.
- [2] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. H. Byers, Big data: The next frontier for innovation, competition, and productivity, 2011.
- [3] W. E. Forum, Big data, big impact: New possibilities for international development, http://www.weforum.org/docs/WEF_TC_MFS_BigData_BigImpact_Briefing_2012.pdf (2012).
- [4] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, Y. Huang, T-drive: driving directions based on taxi trajectories, in: *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, ACM, 2010.
- [5] B. D. Ziebart, A. L. Maas, A. K. Dey, J. A. Bagnell, Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior, in: *Proceedings of the 10th ACM international conference on Ubiquitous computing*, 2008.
- [6] F. Giannotti, M. Nanni, F. Pinelli, D. Pedreschi, Trajectory pattern mining, in: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.
- [7] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma, Mining interesting locations and travel sequences from gps trajectories, in: *Proceedings of the 18th ACM international conference on World wide web*, 2009.
- [8] L. Liu, C. Andris, C. Ratti, Uncovering cabdrivers behavior patterns from their digital traces, *Computers, Environment and Urban Systems* 34 (6) (2010) 541-548.
- [9] X. Wang, K. T. Ma, G.-W. Ng, W. E. L. Grimson, Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models, *International journal of computer vision* 95 (3) (2011) 287-312.
- [10] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, S. Li, ibat: detecting anomalous taxi trajectories from gps traces, in: *Proceedings of the 13th ACM international conference on Ubiquitous computing*, 2011.
- [11] C. Chen, D. Zhang, P. S. Castro, N. Li, L. Sun, S. Li, Z. Wang, iboat: Isolation-based online anomalous trajectory detection, *Intelligent Transportation Systems, IEEE Transactions on* 14 (2) (2013) 806-818.
- [12] Y. Ge, H. Xiong, C. Liu, Z.-H. Zhou, A taxi driving fraud detection system, in: *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, 2011.
- [13] Y. Ge, C. Liu, H. Xiong, J. Chen, A taxi business intelligence system, in: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011.
- [14] E. M. Knox, R. T. Ng, Algorithms for mining distance based outliers in large datasets, in: *Proceedings of the International Conference on Very Large Data Bases*, Citeseer, 1998.
- [15] E. M. Knorr, R. T. Ng, Finding intensional knowledge of distance-based outliers, in: *VLDB, Vol. 99*, 1999, pp. 211-222.
- [16] E. M. Knorr, R. T. Ng, V. Tucakov, Distance-based outliers: algorithms and applications, *The VLDB Journal The International Journal on Very Large Data Bases* 8 (3-4) (2000) 237-253.
- [17] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, in: *ACM SIGMOD Record*, Vol. 29, 2000.
- [18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: identifying density-based local outliers, in: *ACM Sigmod Record*, Vol. 29, 2000.
- [19] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, C. Faloutsos, Loci: Fast outlier detection using the local correlation integral, in: *Data Engineering, 2003. IEEE 19th International Conference on*, 2003.
- [20] V. Barnett, T. Lewis, *Outliers in statistical data*, Vol. 3, Wiley New York, 1994.
- [21] C. C. Aggarwal, P. S. Yu, Outlier detection for high dimensional data, in: *ACM Sigmod Record*, Vol. 30, 2001.
- [22] Z. Liao, Y. Yu, B. Chen, Anomaly detection in gps data based on visual analytics, in: *Visual Analytics Science and Technology (VAST), IEEE Symposium on*, 2010.
- [23] X. Li, J. Han, S. Kim, H. Gonzalez, Roam: Rule-and motif-based anomaly detection in massive moving object data sets., in: *SDM*, Vol. 7, SIAM, 2007, pp. 273-284.
- [24] R. R. Sillito, R. B. Fisher, Semi-supervised learning for anomalous trajectory detection., in: *BMVC*, 2008, pp. 1-10.
- [25] P. S. Castro, D. Zhang, C. Chen, S. Li, G. Pan, From taxi gps traces to social and community dynamics: A survey, *ACM Computing Surveys (CSUR)* 46 (2) (2013) 17.
- [26] Wikipedia, Openstreetmap, <http://en.wikipedia.org/wiki/OpenStreetMap>, accessed: 2015-01-02.
- [27] M. Haklay, P. Weber, Openstreetmap: User-generated street maps, *Pervasive Computing*, IEEE 7 (4) (2008) 12-18.
- [28] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, Y. Huang, Map-matching for low-sampling-rate gps trajectories, in: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, 2009, pp. 352-361.
- [29] L. Bergroth, H. Hakonen, T. Raita, A survey of longest common subsequence algorithms, in: *String Processing and Information Retrieval, IEEE Seventh International Symposium on*, 2000, pp. 39-48.
- [30] D. Eppstein, Finding the k shortest paths, *SIAM Journal on computing* 28 (2) (1998) 652-673.
- [31] J. Y. Yen, Finding the k shortest loopless paths in a network, *management Science* 17 (11) (1971) 712-716.
- [32] M. Piorkowski, N. Sarafijanovic-Djukic, M. Grossglauser, CRAW-DAD data set epfl/mobility (v. 2009-02-24), Downloaded from <http://crawdad.org/epfl/mobility/> (Feb. 2009).
- [33] Y. Bu, L. Chen, A. W.-C. Fu, D. Liu, Efficient anomaly monitoring over moving object trajectory streams, in: *Proceedings of the 15th ACM*

SIGKDD international conference on Knowledge discovery and data mining, 2009.

- [34] Y. Ge, H. Xiong, Z.-h. Zhou, H. Ozdemir, J. Yu, K. C. Lee, Top-eye: top-k evolving trajectory outlier detection, in: Proceedings of the 19th ACM international conference on Information and knowledge management, 2010.
- [35] J.-G. Lee, J. Han, X. Li, Trajectory outlier detection: A partition-and-detect framework, in: Data Engineering, IEEE 24th International Conference on, 2008.
- [36] X. Li, Z. Li, J. Han, J.-G. Lee, Temporal outlier detection in vehicle traffic data, in: Data Engineering, IEEE 25th International Conference on, 2009