Detecting Unmetered Taxi Rides from Trajectory Data

Xibo Zhou [1], Ye Ding [2], Fengchao Peng [1], Qiong Luo [1], Lionel M. Ni [3]
[1] *Department of Science and Engineering, The Hong Kong University of Science and Technology*
[2] *Guangzhou HKUST Fok Ying Tung Research Institute, The Hong Kong University of Science and Technology*
[3] *University of Macau*
{xzhouaa,fpengaa,luo}@cse.ust.hk, yeding@ust.hk, ni@umac.mo

***Abstract*—Taxi fraud has become a serious problem in many large cities, where passengers are overcharged by taxi drivers in various ways. Researchers have developed a number of methods to detect taxi frauds with the assumption that fraudulent trips, among normal trips, are recorded by taximeters. In this paper, different from the previous work, we identify a new type of taxi fraud called unmetered taxi rides, where taxi drivers carry passengers without activating the taximeters. Since these fraudulent rides are not recorded by taximeters, previous detection approaches cannot directly apply to them. Hence, we propose a novel fraud detection system specifically designed for unmetered taxi rides. Our system uses a learning model to detect unmetered trajectory segments that are similar to metered rides, and introduces a heuristic algorithm to construct maximum fraudulent trajectories from the trajectory dataset. We have conducted detailed experiments on real-world datasets, and the results show that the proposed system can detect unmetered taxi rides effectively and efficiently.**

***Keywords*-taxi fraud detection; trajectory; anomaly detection**

I. Introduction

In modern cities, taxi service is an important part of the public transportation system, providing convenience for our daily life. However, in recent years, taxi fraud, in terms of overcharging passengers, has become a serious problem in many large cities, which causes complaints from passengers and damages the reputation of taxi service.

There are many forms of taxi frauds, including: 1) detour [1][2], where taxi drivers overcharge passengers by deliberately taking unnecessary detours; 2) taximeter tampering [3], where taxi drivers tamper the taximeters so that they record longer distances than the actual; and 3) passenger denial [4], where taxi drivers refuse to deliver passengers to their destinations due to various reasons. Many approaches are proposed to detect these behaviors by utilizing these properties.

In this paper, we identify another type of taxi fraud, namely **unmetered taxi rides**, where taxi drivers carry passengers without activating the taximeters. By having the meter off, taxi drivers can overcharge passengers without being recorded. Unmetered taxi rides are a common and severe problem in large cities with complex traffic situations. They hurt the quality of taxi service, break the consistency of taxi pricing, and are usually difficult to track or regulate by taxi companies because *no meter record is collected for these unmetered trips*. Therefore, it is necessary to develop a method to detect such type of fraud.

Unfortunately, existing approaches are often unsuitable for detecting unmetered taxi rides because they assume that the fraud trips are recorded by taximeters. A possible method is to utilize the occupancy information collected from seat sensors on taxis. However, the occupancy information is inaccurate, which may be caused by aging or deliberate damage of these sensors. For instance, a considerable number of taxis that have metered records show vacant occupancy status.

In this paper, we propose a novel taxi fraud detection system specifically for unmetered taxi rides. The system finds anomalous trajectories that are not recorded by the taximeter but have driving behaviors similar to regular metered trips by utilizing both the taxi trajectory data and the taximeter records. The contributions of this paper lie in the following aspects:

- We introduce a new type of taxi fraud called **unmetered taxi rides**. Different from previous taxi frauds, we aim to find anomalous taxi trajectories that are similar to metered trips but not recorded by taximeters.
- We propose a learning model to predict the passenger occupancy status of taxis from unmetered trajectories, and implement a heuristic algorithm to construct anomalous unmetered trajectories.
- We evaluate our system on real-world taxi trajectory data. The results show that our system is effective to find unmetered taxi rides.

II. Preliminaries

A. Problem Definition

Definition 1 (Tracing Record): A tracing record r of a taxi is denoted as a tuple $r = \langle id, t, p, o \rangle$, where $r(id)$ is the taxi id, $r(t)$ is the record time, $r(p)$ is the location point of the taxi at $r(t)$, and $r(o)$ is the occupancy status of the taxi at $r(t)$.

A location point is represented by its latitude and longitude. The occupancy status is a boolean value, recorded as 1 if the taxi is **occupied** by a passenger and 0 if the taxi is **vacant**.

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

Definition 2 (Taximeter Record): A taximeter record m of a metered trip is denoted as a tuple $m = \langle id, st, et \rangle$, where $m(id)$ is the taxi id, $m(st)$ is the start time of the metered trip, and $m(et)$ is the end time of the metered trip.

Given a tracing record r and a set of taximeter records M, we say r is **metered** if $\exists m \in M$ where $r(id) = m(id)$ and $m(st) \leq r(t) \leq m(et)$, and **unmetered** otherwise.

Definition 3 (Trajectory): A trajectory l of a taxi with id tid is a sequence of tracing records denoted as $l = (r_1, r_2, \cdots, r_n)$, where $r_i(id) = tid$ for $i = 1, \cdots, n$. We denote $r_i \in l$ for $i = 1, \cdots, n$ and $|l| = n$.

For simplicity, we denote the taxi id of trajectory l as $l(id)$, where $l(id) = r_i(id)\ \forall r_i \in l$. Given the trajectory l, a trajectory $l' = (r_s, r_{s+1}, \cdots, r_d)$ of consecutive tracing records where $1 \leq s < d \leq |l|$ is called a sub-trajectory of $l = (r_1, r_2, \cdots, r_n)$, denoted as $l' \subseteq l$.

Definition 4 (Fraud Trajectory): Given a trajectory l, we say l is a *fraud trajectory* if r is unmetered and occupied $\forall r \in l$.

Since the occupancy status contained in real-world trajectory datasets is usually imprecise, we need to predict the occupancy for each tracing record before detecting fraud trajectories.

Definition 5 (Occupancy Prediction Problem): Given a set of trajectories L, for each tracing record $r \in l$ where $l \in L$, predict the value of $r(o)$.

After detecting the occupied tracing records, our next problem is to find the maximum fraud trajectories from the unmetered trajectory set.

Definition 6 (Maximum Fraud Trajectory Problem): Given a set of trajectories L, for each trajectory $l \in L$, find a *maximum fraud trajectory* l' where: 1) $l' \subseteq l$, 2) l' is a fraud trajectory, and 3) $|l'| \geq |l''|$, $\forall l'' \subseteq l$ where l'' is a fraud trajectory.

In this paper, we solve both Problems 5 and 6.

B. Overview

Figure 1 shows the workflow of our taxi fraud detection system. In the pre-processing phase, raw trajectories are segmented into metered and unmetered trajectories based on taximeter records. The metered trajectories are filtered from the trajectory set, and the unmetered trajectories are then map-matched into connected paths constrained to the road network. In the feature extraction phase, the map-matched unmetered trajectories are first fragmented into unit segments with constant length. Then, a vector of features are calculated for each unit segment, including spatial-temporal features (such as average travel speed and arc-chord ratio) and statistical features (such as transition frequency between roads and cheating occurance of each taxi id). In the anomaly detection phase, given the feature vector set extracted from fragmented unmetered unit segments, we implement an integrated predictor to detect the occupied instances. After detecting the unit segments that are occupied but not metered, a trajectory construction algorithm is performed to heuristically form the longest fraud trajectories.

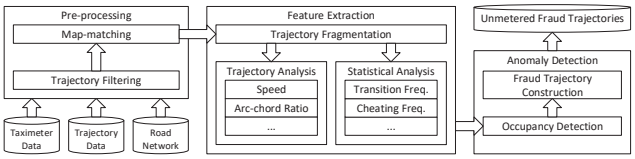


Figure 1: The workflow of taxi fraud detection.

III. Pre-processing

A. Trajectory Filtering

Since we assume the taximeter records are accurate, the first task is to obtain all the unmetered trajectories from the trajectory set. Due to the difference of data formats for these two dataset, we implement a matching method to segment the trajectory dataset into metered and unmetered trajectories. First, we sort all the taximeter records by taxi ids. For each taxi id, we sort its taximeter records by start time. Then, for each taxi id, we can obtain all the metered and unmetered time periods. More specifically, for a sorted taximeter records of taxi id tid denoted as $(tid, st_1, et_1), (tid, st_2, et_2), \cdots, (tid, st_n, et_n)$, the metered time periods are $(st_1, et_1), (st_2, et_2), \cdots, (st_n, et_n)$, whereas the unmetered time periods are (et_1, st_2), $(et_2, st_3), \cdots, (et_{n-1}, st_n)$. Next, we obtain all the trajectories for each taxi id, and locate each tracing record of the trajectories into the matching time period. More specifically, given a taxi id tid with its metered and unmetered time periods obtained, and a trajectory l where $l(id) = tid$, the matching time period for $r \in l$ is $(st_i, et_i), st_i \leq r(t) \leq et_i$ or $(et_i, st_{i+1}), et_i \leq r(t) \leq st_{i+1}$. Finally, for each taxi id and each of its unmetered time period, we fetch all the tracing records located in it, and construct an unmetered trajectory by sorting these records by time.

B. Map-matching

In practice, the location information of raw trajectories are usually imprecise due to the measurement noise and sampling errors. In order to extract spatial features from these trajectories, we perform a map-matching task by aligning the location points to the road networks. We use the Hidden Markov Model (HMM)-based map-matching algorithm [5]. The algorithm builds a hidden Markov model for each map-matching task, where a trajectory represents an observation sequence, each location point is an observation, and each candidate road segment is a hidden state. The algorithm defines the emission and transition probabilities based on distance information, and uses Viterbi algorithm to compute the best path, which gives an inference of the correct road segment for each location point. Each two consecutive match points are connected by the shortest path between them along the road network, which guarantees the connectivity of the entire path.

IV. Feature Extraction

A. Trajectory Fragmentation

Given the set of unmetered trajectories, before we start to extract features, it is necessary to fragment these trajectories into small pieces. This is because the lengths of unmetered trajectories are significantly varied, and each unmetered trajectory might be mixed with occupied and vacant tracing records. We fragment the unmetered trajectories into unit segments with a constant length, and extract the corresponding map-matched path for each unit segment. In the following steps of our system, the feature extraction, occupancy detection and fraud trajectory construction are based on unit segments rather than tracing records.

B. Feature Analysis

After fragmenting the unmetered trajectories into unit segments, we extract features that have impact on occupancy. In general, there are two types of features extracted by our system, namely spatial-temporal features and statistical features. The spatial-temporal features reflect the individual pattern of each trajectory, including average speed, tortuosity, arc-chord and time; and the statistical features reflect the statistical information of the trajectories, including road transition frequency and taxi cheating frequency. We introduce these features briefly:

1) Average Speed: Given a unit segment $u = (r_i, \cdots, r_j) \subset l$, the average speed v_{avg} of u is:

$$v_{avg} = \frac{\text{rdist}(p'_i, p'_j)}{r_j(t) - r_i(t)} \tag{1}$$

where p'_i is the match point of r_i, p'_j is the match point of r_j, and $\text{rdist}(p'_i, p'_j)$ is the driving distance along the map-matched path P' of u from p'_i to p'_j.

2) Arc-chord Ratio: Given a unit segment $u = (r_i, \cdots, r_j) \subset l$, the arc-chord ratio τ_a of u is:

$$\tau_a = \frac{\text{rdist}(p'_i, p'_j)}{\text{cdist}(p'_i, p'_j)} \tag{2}$$

where p'_i is the match point of r_i, p'_j is the match point of r_j, $\text{rdist}(p'_i, p'_j)$ is the driving distance along the map-matched path P' of u from p'_i to p'_j, and $\text{cdist}(p'_i, p'_j)$ is the great circle distance between p'_i and p'_j.

3) Curvature: As described in section III-B, the map-matched path is a sequence of road segments, thus the path from p'_i to p'_j can be represented as a polyline $(d_1, d_2, \cdots, d_m)$ where each d_i represents a line segment. The curvature τ_c of $(d_1, d_2, \cdots, d_m)$ is:

$$\tau_c = \frac{\sum_{k=1}^{m-1} \left(\frac{\angle(d_k, d_{k+1}) \times \pi}{180} \times \frac{1}{\text{cdist}(d_k)}\right)^2}{\text{cdist}(p'_i, p'_j)} \tag{3}$$

where $\angle(d_k, d_{k+1})$ is the intersection angle of d_k and d_{k+1} and $\text{cdist}(d_k)$ is the length of d_k.

4) Time: We first split a day into a number of slots μ_t with a constant period, and then flatten each spatial-temporal feature into a vector based on the corresponding time slot. More specifically, given a unit segment $u = (r_i, \cdots, r_j)$ and a constant time period Δ_t, the index h of the corresponding time slot for u is:

$$h = \frac{\max(r_i(t), r_j(t))}{\Delta_t} \tag{4}$$

For each spatial-temporal feature f extracted from u, it is flattened as $f_{flat} = \langle \underbrace{0, \cdots, 0}_{h}, f, \underbrace{0, \cdots, 0}_{\mu_t - h - 1} \rangle$, where μ_t is the total number of time slots.

5) Road transition frequency: Given a map-matched path $P = (e_1, e_2, \cdots, e_n)$, we say e_{i+1} is a **follower** of e_i in P. Given a set of map-matched path P_{set} and a road segment e, we can obtain a set of followers E_f of e where $\forall e_f \in E_f$, e_f is a follower of e in at least one path in P_{set}. For each follower $e_f \in E_f$, the road transition frequency from e to e_f is $Fr(e, e_f) = \mu_{e_f} / \mu_{E_f}$, where μ_{e_f} is the number of times when e_f is a follower of e in a path $P \in P_{set}$, and μ_{E_f} is the number of times when e has a follower in a path $P \in P_{set}$. Given a unit segment $u = (r_i, \cdots, r_j)$ along with its map-matched path $P' = (e_s, e_{s+1}, \cdots, e_t)$, the road transition frequency of u is:

$$Fr(u) = \frac{\sum_{k=s}^{t-1} Fr(e_k, e_{k+1})}{t - s + 1} \tag{5}$$

In order to distinguish the road transition frequency for occupied and vacant unit segments, we measure the road transition frequency on P_{set} of occupied and vacant unit segments, respectively. The road transition frequency for occupied and vacant unit segments are denoted as $Fr_o(u)$ and $Fr_v(u)$, receptively.

6) Taxi cheating frequency: Given a taxi id tid and a set of its unmetered unit segments, the taxi cheating frequency of tid is:

$$Fr(tid) = \frac{\mu_o}{\mu_v} \tag{6}$$

where μ_o is the number of occupied unit segments, and μ_v is the number of vacant unit segments.

V. Anomaly Detection

A. Occupancy Detection

In this paper, we develop an integrated predictor by utilizing the features introduced in Section IV-B to detect the occupied segments from the set of fragmented unmetered unit segments. First, we build a stochastic gradient descent (SGD) model on each transformed spatial-temporal feature to make a preliminary prediction of occupancy. Then, we integrate the normalized likelihoods of the prediction results

from each SGD model. Formally, given an instance u and a threshold λ, the occupancy of u is:

$$u(o) = \begin{cases} 1 & \frac{1}{n}\sum_{k=1}^{n} \mathscr{L}_i \geq \lambda \\ 0 & otherwise \end{cases} \quad (7)$$

where n is the number of utilized spatial-temporal features, and $\mathscr{L}_i$ is the likelihood of prediction result using the i-th feature.

In our dataset, the percentage of occupied unmetered unit segments out of the entire unmetered unit segments is less than 5%. With such a skewed label distribution, it is easy for a classification model to ignore the occupied labels and predict all the instances as vacant. In order to solve this problem, we implement a *randomized training process*. We first divide the training set into occupied and vacant feature sets, and extract statistical features on the entire occupied and vacant sets, respectively. Then, we evenly partition the occupied and vacant set into n_p groups, respectively. Since there are only 5% occupied segments, the size of each vacant group is 19 times as large as the size of each occupied group. We further split each vacant group into 19 piles with equal sizes. Thus, the entire training set is partitioned into n_p occupied and $19 \times n_p$ vacant piles. Next, in the training phase, each time we randomly pick one occupied and one vacant pile to form a temporal training set, build the classification model, and use it to predict labels of the remaining occupied and vacant groups. Since there are $19 \times n_p \times n_p$ combinations to form the temporal training set, it is time-consuming to validate all the combinations. To deal with this problem, we stop the cross-validation as long as all the occupied set has been used at least once. Finally, we use the trained model to predict the occupancy labels of the test set.

B. Maximum Fraud Trajectory Construction

Since the occupied unit segments detected using our model are discrete, we implement a heuristic method to construct complete occupied unmetered sub-trajectories. Given a sequence of unit segments U fragmented from trajectory l, a set of unit segments $U_t \subset U$ where each $u \in U_t$ is detected to be occupied, and a constant γ, our sub-trajectory construction algorithm is shown in Algorithm 1.

The algorithm starts from each detected occupied unit segment $u \in U_t$, and heuristically searches its preceding $(\S5)$ and following $(\S10)$ unit segment in order, respectively. For each target unit segment, if it is initially detected as vacant, we utilize the road transition frequency to infer the confidence of it being occupied $(\S6, 11)$. More specifically, given a unit segment u, the confidence of u being occupied is:

$$S(u) = (1 + Fr_o(u) - Fr_v(u)) \times \mathscr{L}(u) \quad (8)$$

Algorithm 1 Maximum Fraud Trajectory Construction.

Input: The sequence of unit segments $U = (u_1, \cdots, u_n)$; the set of occupied unit segments $U_t = \{u_{i_1}, u_{i_2}, \cdots, u_{i_k}\}$ where $1 \leq i_1 < i_2 < \cdots < i_k \leq n$; constant ϵ and γ

Output: The set of occupied trajectories L_t

1: $L_t = \emptyset$; $U_o = [1$ if $u_i \in U_t$ else 0 for $u_i \in U]$
2: **for** $j = 1$; $j \leq k$; $j++$ **do**
3: $\quad i_s = i_{j-1} + 1$
4: $\quad i_t = i_{j+1} - 1$
5: $\quad$ **for** $q = i_j - 1$; $q \geq i_s$; $q--$ **do**
6: $\quad\quad$ **if** $U_o[q] = 0$ and $S(u_q) \geq \epsilon$ **then**
7: $\quad\quad\quad U_o[q] = 1$
8: $\quad\quad$ **else**
9: $\quad\quad\quad$ break
10: $\quad$ **for** $q = i_j + 1$; $q \leq i_t$; $q++$ **do**
11: $\quad\quad$ **if** $U_o[q] = 0$ and $S(u_q) \geq \epsilon$ **then**
12: $\quad\quad\quad U_o[q] = 1$
13: $\quad\quad$ **else**
14: $\quad\quad\quad$ break
15: $l = \emptyset$
16: **for** $j = 1$; $j \leq n$; $j++$ **do**
17: $\quad$ **if** $U_o[q] = 1$ **then**
18: $\quad\quad l = l + \{u_j\}$
19: $\quad$ **else**
20: $\quad\quad$ **if** $|l| \geq \gamma$ **then**
21: $\quad\quad\quad L_t = L_t + \{l\}$
22: $\quad\quad\quad l = \emptyset$
23: **if** $|l| \geq \gamma$ **then**
24: $\quad L_t = L_t + \{l\}$
25: **return** L_t

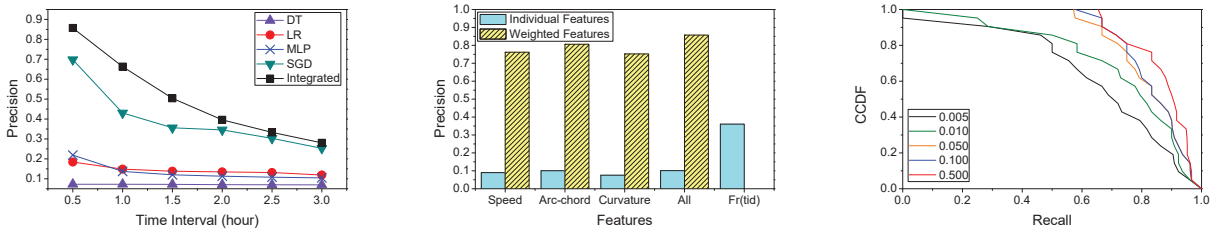
where $\mathscr{L}(u)$ is the likelihood of occupancy prediction result using our model. If $S(u)$ exceeds a specified threshold ϵ, it is inferred as occupied. If so, we continue the heuristic search. The process terminates when the next unit segment is already detected as occupied, or it is still inferred as vacant $(\S8 - 9, \S13 - 14)$. After checking all the detected occupied unit segment $u \in U_t$, the algorithm connects all the consecutive unit segments that are detected or inferred as occupied $(\S15 - 24)$. For each connected unit segment, if it is longer than the specified threshold γ $(\S20, \S23)$, a maximum fraud trajectory is constructed $(\S21, \S24)$.

VI. Evaluation

A. Experiment Setup

1) Experiment Environment: The experiments are conducted on a server with Intel Core i5-4590 CPU and 16 GB RAM. The operating system is Ubuntu 14.04, and the code is written in Python 2.7.6.

2) Real World Dataset: We use a dataset collected from a large city in China. The dataset contains 154 million taxi



(a) Precision under different Δ_t (b) Precision using different features (c) CCDF curves of recall under different γ

Figure 2: Effectiveness of integrated predictor and construction algorithm.

tracking records and 4 million taximeter records of 15,231 taxis for 26 days [6]. 69% of the taxi tracking records are unmetered, and 5% of the unmetered taxi tracking records are occupied. The road network used for map-matching consists of 25,613 intersections and 36,451 road segments. Since the occupancy status is inaccurate, we use the information of metered trajectories to select reliable occupancy labels. Given a set of metered trajectories L and a taxi id tid, the reliability on occupancy of tid is defined as $R(tid) = \frac{\mu_{occupy}}{\mu_{all}}$, where μ_{occupy} is the number of tracing records $r_i \in L$ where $r_i(id) = tid$ and $r_i(o) = 1$, and μ_{all} is the number of tracing records $r_i \in L$ where $r_i(id) = tid$. We use the trajectories of taxis with reliability greater than 90% as ground truth.

3) Evaluation Metrics: We use *precision* to evaluate the effectiveness of our occupancy detection model, which is denoted as $precision = \frac{\mu_{tp}}{\mu_p}$, where μ_{tp} is the number of occupied unit segments that are correctly detected, and μ_p is the total number of unit segments detected as occupied.

We use *recall* to evaluate the effectiveness of our maximum fraud trajectory construction algorithm. Given a fraudulent trajectory $l' \subset l$ and the corresponding maximum fraud trajectory l'' constructed by our algorithm, the *recall* of l'' is defined as $recall = \frac{|l' \bigcap l''|}{|l'|}$, where $|l' \bigcap l''|$ is the number of unit segments contained in the intersection sub-trajectory of l' and l''.

B. Experiment Results

In our experiments, we use the first thirteen days of taxi tracking records and taximeter records as training set, and the rest as test set. We partition the training set using $n_p = 10$. In the occupancy detection phase, the features we use includes average speed, arc-chord, curvature, and taxi cheating frequency. As existing approaches on anomaly detection [7][8][9][10][11][12][13] are unsuitable for detecting unmetered taxi rides, the prediction models we use for comparison include logistic regression (LR), stochastic gradient descent (SGD), decision tree (DT), and multi-layer perceptron (MLP). The default values of parameters are: $w = 1$, $\Delta_t = 30$ min, $\lambda = 0.5$, and $\gamma = 0.5$.

First, we compare the effectiveness of our integrated predictor with the baseline prediction models under different granularity of time interval for feature flattening. The results are shown in Figure 2(a). Our integrated predictor always achieves a much higher positive predictive value than the baseline prediction models under different Δ_t, which also shows the consistency of our integrated predictor. In particular, the positive predictive value of our integrated predictor is up to 85% when Δ_t is 30 minutes, whereas some of the baseline prediction models result in less than 30%. Moreover, the positive predictive value of occupancy detection decreases when the time Δ_t increases for all the prediction models. This result shows the importance of time for feature flattening.

Next, we further analyze the performance of our selected features for occupancy detection. We conduct experiments on our integrated predictor by utilizing: 1) each spatial-temporal feature individually, compared with all features together; and 2) each spatial-temporal feature multiplied by taxi cheating frequency as a weight (referred to as transformed features), compared with taxi cheating frequency as an individual feature. All the features are flattened by time. The results are shown in Figure 2(b). The performance of spatial-temporal features without considering taxi cheating frequency is low, and taxi cheating frequency is not effective as an independent feature. However, the positive predictive values of occupancy detection rise significantly if we treat taxi cheating frequency as a weight and multiply it with each spatial-temporal feature. Finally, our integrated predictor achieves better performance than only considering each individual one by taking all the features into modeling.

Last, we evaluate the effectiveness of our trajectory construction algorithm with γ varied from 0.005 to 0.5. For the fraudulent trajectory set constructed by each group of experiments, we use the complementary cumulative probability function (CCDF) curve to evaluate its recall distribution. The results are shown in Figure 2(c). In general, the CCDF curve of the recall distribution leans towards 100% when γ increases. After γ reaches 0.5, the trajectory construction algorithm achieves the best performance, and the CCDF curve stays unchanged. Moreover, the results show that 81% of the trajectories constructed by our algorithm achieve over 75% recall, which indicates the effectiveness of our algorithm.

(a) Case 1 (b) Case 2

Figure 3: Examples illustrating the effectiveness of our taxi fraud detection system.

C. Case Study

We conduct several case studies to exhibit the effectiveness of our taxi fraud detection system. Figure 3 illustrates two identical examples of the anomalous trajectories detected by our system. In Figure 3(a), the taxi is driving from the high-speed railway station to a suburban residential area. In Figure 3(b), the taxi is driving from the port of entry to an urban office area. Based on our investigations, these trajectories are highly suspicious to be unmetered taxi rides because of three reasons. First, these departure areas are important traffic terminals. For most of the time, there are crowds of people waiting in line to hire taxis, and regular taxis also have to wait in line to pick up passengers. Due to the inconvenience caused by congestion, it is not likely for a vacant taxi to travel into these areas. Second, there are huge demands of taxis in these departure areas, especially for those citizens traveling between urban and suburban areas every day. Since the profit of picking up passengers in these areas are quite high, it is not likely for a taxi to leave for urban or suburban areas without carrying passengers. Third, these trajectories mainly travel through expressways or main roads. It is efficient and fast to choose these routes for an occupied taxi to deliver passengers, but not effective for a vacant taxi to hunt for passengers. Hence, it is not likely for a vacant taxi to travel along these trajectories. In conclusion, these examples demonstrate that our system is effective in finding unmetered taxi rides with convincing evidences.

VII. Conclusion

In this paper, we study a new type of taxi fraud called **unmetered taxi fraud**, where taxi drivers carry passengers without activating the taximeters. Existing approaches are not suitable for our problem because: 1) unmetered fraudulent trips are not recorded by taximeters, and 2) the occupancy information collected from seat sensors on taxis is inaccurate. Therefore, we propose a novel taxi fraud detection system specifically for unmetered taxi frauds. The basic idea of our approach is to find anomalous trajectories that are not recorded by the taximeter but have driving behaviors similar to regular occupied trips. We first propose a learning model to capture the different driving behaviors of taxis when they are occupied or vacant, and then implement a heuristic algorithm to construct unmetered fraudulent trajectories by utilizing trajectory dataset and taximeter records. We conduct intensive experiments on real trajectory data. The results show that our proposed system achieves a satisfactory performance.

Acknowledgments

This work is supported in part by the National Key Basic Research and Development Program of China (973) Grant 2014CB340304, Macao FDCT Grant 149/2016/A and the University of Macau Grant SRG2015-00050-FST.

References

[1] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "ibat: detecting anomalous taxi trajectories from gps traces," in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 99–108.

[2] Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou, "A taxi driving fraud detection system," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 181–190.

[3] S. Liu, L. M. Ni, and R. Krishnan, "Fraud detection from taxis' driving behaviors," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 464–472, 2014.

[4] S. Zhang and Z. Wang, "Inferring passenger denial behavior of taxi drivers from large-scale taxi traces," *PloS one*, vol. 11, no. 11, p. e0165597, 2016.

[5] X. Zhou, Y. Ding, H. Tan, Q. Luo, and L. M. Ni, "Himm: An hmm-based interactive map-matching system," *Lecture Notes in Computer Science*, p. 3, 2017.

[6] Y. Ding, S. Liu, J. Pu, and L. M. Ni, "Hunts: A trajectory recommendation system for effective and efficient hunting of taxi passengers," in *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, vol. 1. IEEE, 2013, pp. 107–116.

[7] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE, 2008, pp. 140–149.

[8] Y. Bu, L. Chen, A. W.-C. Fu, and 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*. ACM, 2009, pp. 159–168.

[9] X. Li, Z. Li, J. Han, and J.-G. Lee, "Temporal outlier detection in vehicle traffic data," in *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*. IEEE, 2009, pp. 1319–1322.

[10] Y. Ge, H. Xiong, Z.-h. Zhou, H. Ozdemir, J. Yu, and K. C. Lee, "Top-eye: Top-k evolving trajectory outlier detection," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 1733–1736.

[11] C. Chen, D. Zhang, P. S. Castro, N. Li, L. Sun, and S. Li, "Real-time detection of anomalous taxi trajectories from gps traces," in *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, 2011, pp. 63–74.

[12] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Time-dependent popular routes based trajectory outlier detection," in *International Conference on Web Information Systems Engineering*. Springer, 2015, pp. 16–30.

[13] Z. Lv, J. Xu, P. Zhao, G. Liu, L. Zhao, and X. Zhou, "Outlier trajectory detection: A trajectory analytics based approach," in *International Conference on Database Systems for Advanced Applications*. Springer, 2017, pp. 231–246.