FraudTrip: Taxi Fraudulent Trip Detection From Corresponding Trajectories

Ye Ding, *Member, IEEE*, Wenyi Zhang, *Member, IEEE*, Xibo Zhou, *Member, IEEE*, Qing Liao, *Member, IEEE*, Qiong Luo, *Member, IEEE*, and Lionel M. Ni, *Life Fellow, IEEE*

***Abstract*—A passenger is overcharged by the taxi driver is one common type of fraudulent trip, and it brings negative impacts to modern cities. Most existing fraudulent trip detection works rely on the assumption that the trip is correctly recorded by the taximeter. However, there are many taxi drivers in China carrying passengers without activating the taximeter, especially when the taxi driver is trying to overcharge the passengers. Hence, existing detection methods cannot be directly applied to such real-world scenario. In this article, we propose a system, called "FraudTrip," which detects "unmetered" taxi trips based on a novel fraud detection algorithm and a heuristic maximum fraudulent trajectory construction algorithm. Based on the experiments on both synthetic and real-world trajectory data sets, FraudTrip can effectively and efficiently detect fraudulent trips without the help of taximeters.**

***Index Terms*—Anomaly detection, taxi fraud detection, trajectory.**

I. Introduction

TAXI services in modern cities are often corrupted by frauds, and passengers are often overcharged by taxi drivers [1]. For example, in Fig. 1, there are many taxi drivers in Lam Kui Fong (a famous night club center in Hong Kong) try to overcharge the passengers due to that it is: 1) difficult to find public transportation services in the midnight and 2) prohibited to drive after drinking. These taxi frauds result in many complaints and may lead to bad reputation of taxi services.

Common taxi frauds include: 1) taximeter tampering, where the taximeter of a taxi is modified so that the shown driving distance is longer than true driving distance; 2) detour, where the taxi takes an irregular route to deliver a passenger and often takes more driving time and distance than usual; and 3) refusing of service, where the driver refuses to carry a specific passenger, or tries to find a passenger who is willing to pay more fare. These fraudulent behaviors usually have evident properties that differ from normal taxi trips. For example, the distance of a detour is usually longer than normal paths between a pair of source and destination [2]; the reported speed of the taxi with a tampered taximeter tends to be higher than its actual speed [3]; and the taxi drivers with a higher income are more likely to refuse passengers traveling to unpopular areas [4]. There are many existing methods try to detect these types behaviors.

We propose a new type of taxi fraud called *unmetered taxi trip* as carrying passengers without activating the taximeter. In this way, taxi drivers could "negotiate" and often overcharge passengers with a higher fare than usual. Unmetered taxi trip is a serious problem in modern cities. For example, 146 taxi drivers were caught for illegally refusing or arbitrarily charging the passengers within only three days in Guangzhou, China [6]. Unmetered taxi trips are problematic to the society due to three major reasons. First, they hurt the quality of taxi service, especially for tourists that are unfamiliar with the city transportation, or during rush hours when the supply of taxis is short. Second, they lead to an unfair competition between taxis and causes trouble for traffic management. Third, they are difficult to be tracked or regulated due to that there are no taximeter records for the unmetered rides. In conclusion, it is important to design a new fraud detection algorithm for unmetered taxi trips.

However, previous methods are not suitable for the detection of unmetered taxi trips. First, existing approaches usually rely on the taximeters, which is no longer true for unmetered taxi frauds. There are many other ways to find out whether the taxi is occupied or not, such as the information from seat sensors, but such information is not accurate. For example, the sensor cannot distinguish whether the seat is occupied by a passenger or a piece of luggage. Moreover, the sensors of many taxis are not working at all because of ageing or deliberate damages. Based on our observation, 12% of the taxis in our data set have rides with metered records yet the occupancy status showed vacant. Hence, the occupancy information is not reliable and cannot be directly used to detect unmetered taxi frauds. Second, existing approaches assume that fraud trips exhibit anomalous behaviors from normal metered trajectories, which is also not true for unmetered taxi trips. Based on our observation, the driving behavior or dynamics

Manuscript received May 18, 2020; revised July 21, 2020; accepted August 19, 2020. Date of publication August 25, 2020; date of current version August 6, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61976051, Grant 61702134, Grant U19A2067, and Grant U1811463; in part by the National Key Research and Development Program of China under Grant 2017YFB0202201; and in part by the Basic Research Project of Shenzhen under Grant JCYJ20180306174743727. This article was presented in part at the IEEE International Conference on Big Data, Boston, MA, USA, 2017. *(Corresponding author: Qing Liao.)*

Ye Ding and Wenyi Zhang are with the School of Cyperspace Security, Dongguan University of Technology, Dongguan 523808, China (e-mail: dingye@dgut.edu.cn; zwy1754981270@163.com).

Xibo Zhou, Qiong Luo, and Lionel M. Ni are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: xzhouaa@ust.hk; luo@ust.hk; ni@ust.hk).

Qing Liao is with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China (e-mail: liaoqing@hit.edu).

Digital Object Identifier 10.1109/JIOT.2020.3019398

Fig. 1. Taxi driver is trying to overcharge the passengers in Lam Kui Fong, Hong Kong [5].

of unmetered fraud rides are more similar to normal metered trips than to vacant taxi trajectories. Hence, it is unsuitable to perform anomaly detection on metered trajectories. Third, existing approaches treat each metered trajectory as a single object and aim to find anomalous objects. However, an unmetered taxi trip is often a partial segment of an entire unmetered trajectory including when the taxi was vacant. Thus, we aim to detect anomalous trajectory fragments rather than complete trajectory objects. Unfortunately, previous methods are not feasible for this task.

Hence, we build *FraudTrip*, a fraud detection system specifically designed for unmetered taxi trips. First, we propose an occupancy detection algorithm to identify the occupancy of taxis based on a fine-grained set of features generated from taxi trajectories. Second, we apply a maximum fraud trajectory construction algorithm on the predicted occupancy status of taxis. The contributions of this article are as follows.

1) We consider *unmetered taxi trip* in real-world scenarios, which describes the taxi trip that has been recorded as vacant but has similar driving behaviors to regular metered trips.
2) We design a novel system for the detection of unmetered taxi trips, called "FraudTrip," which consists of a learning model which predicts the occupancy status of taxis, and a heuristic algorithm which constructs anomalous unmetered trajectories.
3) We demonstrate the effectiveness and efficiency of FraudTrip on both synthetic and real-world taxi trajectory data sets.

The remainder of this article is organized as follows. We first discuss the related works in Section II and then introduce the problem definitions and the system in Section III. Next, we describe each component of the system in Sections IV–VI, respectively. Finally, we evaluate the system in Section VII and conclude this article in Section VIII.

II. Related Works

A. Anomaly Detection

Existing approaches for general anomaly detection on trajectory data can be categorized into two groups: *spatial anomaly detection* and *temporal anomaly detection*.

For spatial anomaly detection, Lee *et al.* [7] proposed a partition-and-detect framework to detect outlying subtrajectories based on distance and density. Bu *et al.* [8] proposed a window-based approach to detect anomalous subtrajectories inside a window based on distance. Ge *et al.* [9] proposed a spatial anomaly detection framework for evolving trajectories based on travel direction and density. Chen *et al.* [10] focused on detecting trajectories that deviate significantly from histories. Lv *et al.* [11] proposed a prefix tree-based algorithm to calculate the travel frequency of routes and implemented a clustering-based approach to find the anomalous trajectories. Patil *et al.* [12] proposed the GeoSClean, when detected the anomaly points, it considers the combination of properties of the GPS trajectory data as distance, velocity, and acceleration. Zhao *et al.* [13] utilized the background image sequence of videos to implement traffic anomaly detection based on vehicle trajectories and designed a multiobject track (MOT) algorithm suitable for this task. Lu *et al.* [14] proposed a distributed clustering algorithm to solve the problem of massive calculation of distance and neighborhood density in the trajectory anomaly detection algorithm. Ergezer and Leblebicioğlu [15] represented trajectories via co-variance features and its anomaly detection is achieved by sparse representations on nearest neighbors. Wang *et al.* [16] discussed a novel difference and intersection set (DIS) distance metric to evaluate the similarity between any two trajectories and designed an anomaly score function to quantify the differences between different types of anomalous and normal trajectories. Piciarelli *et al.* [17] proposed an approach based on single-class support vector machine (SVM) clustering, where the novelty detection SVM capabilities are used for the identification of anomalous trajectories. Smith *et al.* [18] proposed average p-value as an efficiency criteria for conformal anomaly detection. A comparison with a k-nearest neighbors nonconformity measure is presented and the results are discussed. Lei [19] proposed a framework for maritime trajectory modeling and anomaly detection, called MT-MAD, which takes into account of the fact that anomalous behavior manifests in unusual location points and subtrajectories in the spatial domain as well as in the sequence and manner in which these locations and subtrajectories occur. Kumar *et al.* [20] proposed a novel application of hierarchical clustering algorithms based on visual assessment of tendency (VAT, iVAT, and clusiVAT) for trajectory analysis. Yang *et al.* [21] proposed a TRASMIL framework for local anomaly detection based on trajectory segmentation and multi-instance learning. Sun *et al.* [22] presented a novel GPS-based taxi system which can detect ongoing anomalous passenger delivery behaviors leveraging his proposed iBOAT method. All these approaches are not directly applicable to our problem due to the difference between data sets and the lack of ground truth.

For temporal anomaly detection, Suzuki *et al.* [23] employed a hidden Markov model (HMM) to model time-series features of human positions and a similarity matrix of HMM mutual distances is formed. Li *et al.* [24] tried to identify temporal outliers by utilizing historical similarity trends. Zhu *et al.* [25] proposed a novel algorithm that takes travel time into consideration and detected anomalous trajectories

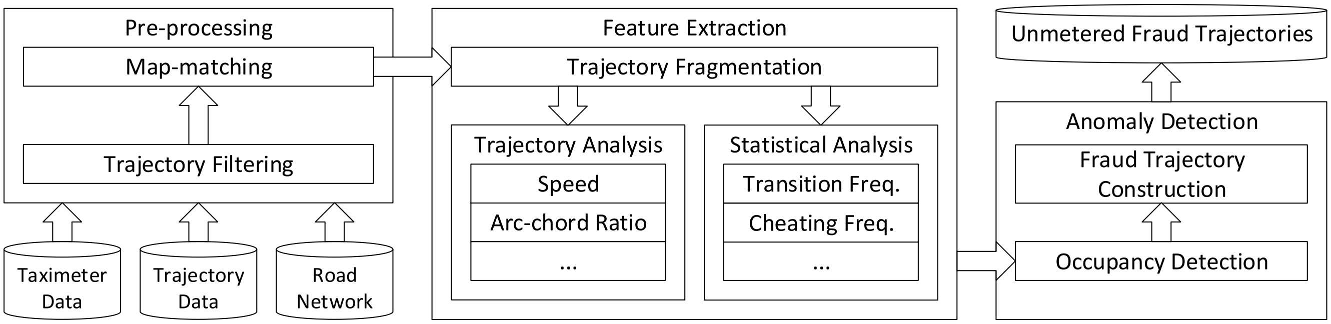


Fig. 2. Workflow of FraudTrip.

based on temporal route popularity. Since taxi frauds are heavily concerned in spatial features, temporal approaches can only act as an enhancement of spatial anomaly detection algorithms, and the performance is often low if we only use temporal algorithms.

B. Taxi Fraud Detection

For taxi fraud detection, Zhang *et al.* [2] and Ge *et al.* [26] introduced a scenario of taxi fraud, where taxi drivers deliberately take unnecessary detours in order to overcharge passengers. They transformed the taxi fraud detection problem into finding anomalous trajectories from all the trajectories with the same source–destination pairs and used the spatial distance as the main feature to design the anomaly detection method. Liu *et al.* [3] introduced another case, where fraud taxi drivers modify the taximeters to a smaller scale so that they record longer distances than the actual ones. They modeled taxi fraud behaviors in a trajectory-free and map-free scenario, constructed a model by the speed information instead of location or distance, and proposed a speed-based clustering method to detect taxi fraud. Different from the above taxi fraud scenarios, we target on the unmetered taxi trips.

III. Preliminaries

In this section, we first formalize the problem of detecting fraud taxi trips, and then introduce the framework of our system.

A. Problem Definition

A *tracing record* is a tuple $r = \langle id, t, p, o\rangle$ consists of taxi ID $r[id]$, record time $r[t]$, location point $r[p]$, and occupancy status $r[o]$, where $r[o] = 0$ if the taxi is *vacant* and $r[o] = 1$ if the taxi is *occupied.* A *taximeter record* is a tuple $m = \langle id, st, et\rangle$ consists of taxi ID $m[id]$, start time $m[st]$, and end time $m[et]$. r is *metered* if $\exists m \in M$ where $r[id] = m[id]$ and $m[st] \le r[t] \le m[et]$, and *unmetered* otherwise. A *trajectory* is defined as $l = (r_1, r_2, \dots, r_n)$ for each taxi. We use $|l|$ to represent the number of tracing records in l.

Hence, an *unmetered taxi trip* is a trajectory where $\forall r \in l$ are both occupied and unmetered. In this article, we consider both the occupancy detection task and the maximum fraud trajectory construction task to jointly detect unmetered taxi trips.

Definition 1 (Occupancy Detection): Predict the occupancy status $r[o]$ for each tracing record $r \in l$, where trajectory $l \in L$ and L is a set of unmetered trajectories.

Definition 2 (Maximum Fraud Trajectory Construction): For each trajectory $l \in L$ and L is a set of unmetered trajectories, find a *maximum fraud trajectory* l' where:

1) $l' \subseteq l$;
2) l' is an unmetered taxi trip;
3) $|l'| \ge |l''|\ \forall l'' \subseteq l$, where l'' is an unmetered taxi trip.

In this article, we will solve Problems 1 and 2 in Sections VI-A and VI-B, respectively.

B. Overview

Fig. 2 shows the workflow of FraudTrip, including three steps.

1) *Preprocessing:* In this step, taxi trajectories are separated into metered and unmetered trajectories and aligned to the road network.
2) *Feature Extraction:* In this step, the spatial–temporal features of each segment of a trajectory are extracted from both metered and unmetered trajectories.
3) *Anomaly Detection:* In this step, a) the actual occupancy status of each segment of a trajectory is predicted through the occupancy detection model and b) a maximum fraud trajectory is constructed from the trajectory segments through the maximum fraud trajectory construction algorithm.

After these three steps, the unmetered parts of an input trajectory will be verified whether they are actually occupied. If so, this part of the original trajectory will be considered as a fraudulent trip.

IV. Preprocessing

In this section, we introduce the data preprocessing phase, including trajectory filtering and map-matching.

A. Trajectory Filtering

According to the system architecture as shown in Fig. 2, in order to study the driving patterns from known metered taxi trips, we have to separate metered and unmetered records from the original trajectory data set. However, the taximeter data set and the trajectory data set are heterogeneous, so we will have to align the data. In order to do so, the taximeter records are first grouped by taxi ID. In each group, the metered and unmetered records are then separated by sorting the records by time and slicing the data when occupancy status changes. More specifically, for a group of sorted taximeter records $(tid, st_1, et_1), (tid, st_2, et_2), \ldots, (tid, st_n, et_n)$ of taxi tid, the corresponding metered time periods are

$$(st_1, et_1), (st_2, et_2), \ldots, (st_n, et_n) \tag{1}$$

and the corresponding unmetered time periods are

$$(et_1, st_2), (et_2, st_3), \ldots, (et_{n-1}, st_n). \tag{2}$$

Given the resulting taximeter records, we can locate the corresponding tracing records and match them together. More specifically, the matching time period for $r \in l$ is

$$(st_i, et_i), st_i \leq r[t] \leq et_i \tag{3}$$

or

$$(et_i, st_{i+1}), et_i \leq r[t] \leq st_{i+1}. \tag{4}$$

Finally, for each taxi and its corresponding metered or unmetered time periods, the corresponding tracing records are extracted to construct an unmetered trajectory.

B. Map-Matching

Map-matching is a calibration technique which aims to align the location points of a trajectory to a road network, thus making the position data accurate for spatial feature extraction. Some previous works on taxi fraud detection try to avoid performing the map-matching task in order to save preprocessing time. For example, Liu *et al.* [3] proposed to calculate the average speed of a metered trajectory by directly using the distance information obtained from its corresponding taximeter record. However, in our problem, it is obvious that the travel distance of unmetered trajectories are not included in taximeter records. Another example [2] tries to approximate the distance information by splitting the city map into grid-cells of equal size, and mapping trajectories into sequences of traversed cells. However, this method is imprecise because the road networks in a city are often distributed unevenly. Therefore, in our system, we perform map-matching on unmetered trajectories before extracting the features.

The definitions and the output format of the map-matching task are as follows.

Definition 3 (Road Segment): A road segment e is a directed polyline between two road intersections v_i and v_j, and there is no other road intersection on e. We denote $v_i \in e$ and $v_j \in e$.

Definition 4 (Road Network): A road network is a weighted directed graph $G = (V, E)$, where V is a set of road intersections (or vertices), and E is a set of road segments (or edges). The weight of a road segment is represented by its properties.

Definition 5 (Path): A path $P = (e_1, e_2, \ldots, e_n)$ is a sequence of road segments where e_i and e_{i+1} are connected for $i = 1, 2, \ldots, n-1$. Two road segments e_i and e_j are connected if there exists some intersection v such that $v \in e_i$ and $v \in e_j$.

Algorithm 1 Greedy Map-Matching Algorithm [27]

```
Require: location point $r[p]$; road network $G$
Ensure: match point $e$ of $r[p]$
split $G$ into a set of equal subspaces $\{g_1, g_2, \ldots, g_n\}$ with
    each $g_i$ contains at least one road segment
find the subspace $g_i$ where $r[p] \in g_i$
$g'_i \leftarrow g_i \cup$ adjacent subspaces of $g_i$
$\Delta \leftarrow \infty$;
for each road segment $e_i \in g'_i$ do
    $d \leftarrow$ minimum perpendicular distance from $r[p]$ to $e_i$
    if $d < \Delta$ then
        $\Delta \leftarrow d$
        $e \leftarrow e_i$
    end if
end for
return $e$
```

Definition 6 (Match Point): Given a tracing record r and a road segment e, we say $r[p]$ is *matched* to e if r was sampled when the taxi was moving on e, and p' is the match point of $r[p]$ on e.

Definition 7 (Map-Matching): Given a trajectory l and a road network G, a map-matching task finds a path P, such that each location point $r_i[p]$ of $r_i \in l$ is matched to exactly one road segment $e_j \in E$. The resulting sequence of match points is denoted as $(p'_1, p'_2, \ldots, p'_n)$.

To solve the above map-matching problem, we could use a simple greedy map-matching as shown in Algorithm 1 [27], or more advanced map-matching algorithms such as HIMM [28].

V. Feature Extraction

In this section, we introduce the feature extraction phase, including trajectory fragmentation and feature computation.

A. Trajectory Fragmentation

The original trajectory should be fragmented due to the following reasons: 1) an unmetered trajectory contains all the tracing records between two continuous metered trips of a taxi, hence the size of an unmetered trajectory is undetermined due to the diversity of occupancy rates and 2) the original trajectory is mixed with occupied and vacant tracing records, hence it is inappropriate to treat the entire trajectory as a single instance. In this article, the fragmentation is based on a constant unit length, and one example is demonstrated in Fig. 3. In this example, trajectory $(r_1, r_2, \ldots, r_7)$ is fragmented into three fragments of constant length 2: (r_1, r_2, r_3), (r_4, r_5, r_6), and (r_7, r_8, r_9). Formally, given an unmetered trajectory l and a unit segment length w, l is fragmented into $\lceil (|l| - 1)/w \rceil$ unit segments, represented as

$$(r_1, \ldots, r_{w+1}), (r_{w+1}, \ldots, r_{2w+1}), \ldots \tag{5}$$

Each unit segment is denoted as $u = (r_i, \ldots, r_j)$, we say $u \subset l$ if u is fragmented from l. Meanwhile, suppose the map-matched path of l is $P = (e_1, e_2, \ldots, e_n)$, then the map-matched path of $u = (r_i, \ldots, r_j)$ is $P' = (e_s, e_{s+1}, \ldots, e_t)$

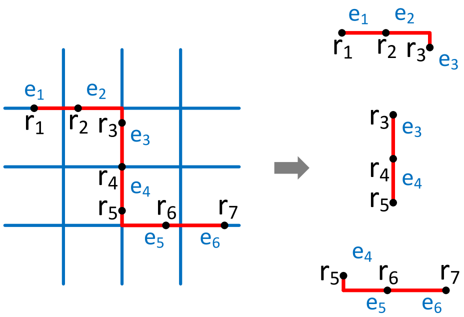


Fig. 3. Example of trajectory fragmentation.

where r_i is matched to e_s and r_j is matched to e_t. We say $u = (r_i, \ldots, r_j)$ is occupied if

$$\frac{\sum_{k=i}^{j} r_k[o]}{j - i + 1} \geq \sigma \tag{6}$$

or vacant otherwise. In this article, we set σ as 0.5.

B. Feature Analysis

With the fragmented unmetered trajectories, we can now extract necessary features for the detection of occupancy status, including spatial–temporal features and statistical features. Spatial–temporal features are widely adopted by traffic prediction works [29], [30], which represent the driving patterns of each trajectory, including average speed, arc-chord ratio, curvature, and time. Statistical features represent the statistical information of the trajectories, including road transition frequency and taxi cheating frequency. The details are shown as follows.

1) Average Speed: Based on our exploration of the trajectory data set, we observe that occupied taxis tend to travel faster than vacant taxis. Therefore, the average speed of each unit segment is an important feature to classify occupancies. Formally, given a unit segment $u = (r_i, \ldots, r_j) \subset l$, the average speed $\bar{v}$ of u is

$$\bar{v} = \frac{r\text{dist}\left(p_i', p_j'\right)}{r_j[t] - r_i[t]} \tag{7}$$

where p_i' is the match point of r_i, p_j' is the match point of r_j, and $r\text{dist}(p_i', p_j')$ is the driving distance along P' of u from p_i' to p_j'.

2) Arc-Chord Ratio: Another observation we find on the trajectory data set is that, occupied taxis usually prefer direct travel routes, while vacant taxis have a high possibility of wandering around. Therefore, the tortuosity of the travel path for each unit segment is also useful to classify occupancies. The simplest mathematical method to estimate tortuosity is the *arc-chord ratio*. More specifically, given a unit segment $u = (r_i, \ldots, r_j) \subset l$, the arc-chord ratio τ_a of u is

$$\tau_a = \frac{r\text{dist}\left(p_i', p_j'\right)}{c\text{dist}\left(p_i', p_j'\right)} \tag{8}$$

where $c\text{dist}(p_i', p_j')$ is the physical distance between p_i' and p_j'.

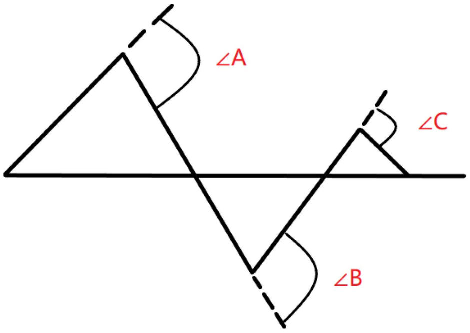


Fig. 4. Example of the curvature of a trajectory.

3) Curvature: Another mathematical method to measure tortuosity is *curvature*. In this article, the "curvature" of a map-matched trajectory is represented as the ratio of the cumulative turning angles of the road segments in the corresponding path and the distance between the source and destination. An example of curvature is shown in Fig. 4, where the curvature of this trajectory is roughly the average of $\angle A$, $\angle B$, and $\angle C$.

Formally, as described in Section IV-B, the path from p_i' to p_j' is a polyline $(d_1, d_2, \ldots, d_m)$, and the curvature τ_c of the path is defined as

$$\tau_c = \frac{\sum_{k=1}^{m-1}\left(\frac{\angle(d_k, d_{k+1}) \times \pi}{180} \times \frac{1}{c\text{dist}(d_k)}\right)^2}{c\text{dist}\left(p_i', p_j'\right)} \tag{9}$$

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

4) Time of Travel: The driving speed and trajectory in the transportation network have the characteristics of period and seasonality. For example, during peak hours or popular sections, taxis may cause changes in driving speed and trajectory due to congestion. Therefore, it is necessary to take the time of travel into consideration during feature extraction. In this article, we flatten each spatial–temporal feature into the corresponding time slot μ_t of each day. Formally, given a time period constant Δ_t and a unit segment $u = (r_i, \ldots, r_j)$, the time slot index h of u is

$$h = \frac{\max\left(r_i[t], r_j[t]\right)}{\Delta_t}. \tag{10}$$

Each spatial–temporal feature f extracted from u is flattened as

$$f = \left\langle \underbrace{0, \ldots, 0}_{h}, f, \underbrace{0, \ldots, 0}_{\mu_t - h - 1} \right\rangle \tag{11}$$

where μ_t is the total number of time slots.

5) Road Transition Frequency: As mentioned above, occupied taxis usually prefer direct travel routes, while vacant taxis have a high possibility of wandering around. Another consequence is that the road transition selections of occupied taxis tend to be more consistent than that of vacant taxis. Therefore, road transition frequency can be used to classify occupancy in some areas of the road network.

Definition 8 (Following Road Segment): Given a path $P = (e_1, e_2, \ldots, e_n)$, we say e_{i+1} is a *follower* of e_i in P for $i = 1, 2, \ldots, n - 1$.

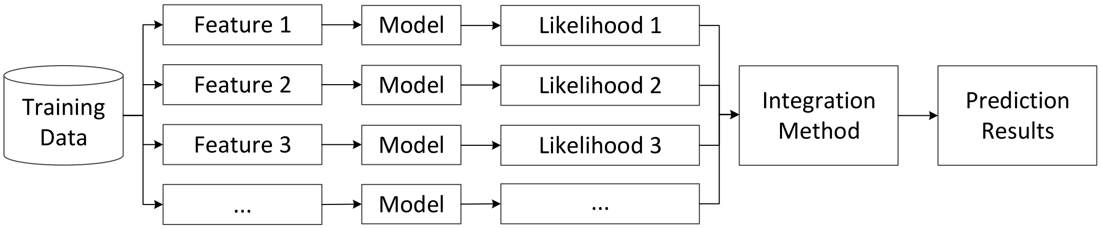


Fig. 5. Structure of our integrated prediction model.

Given a road segment e and a set of paths P_{set}, it is possible to find a set of following road segments E_f corresponding to the paths in P_{set}. Formally, the road transition frequency from e to e_f is defined as:

$$Fr\big(e, e_f\big) = \frac{\mu_{e_f}}{\mu_{E_f}} \tag{12}$$

where μ_{e_f} is the frequency when e_f follows e in a path $P \in P_{\text{set}}$, and μ_{E_f} is the frequency when e has a follower in the path $P \in P_{\text{set}}$. By counting occupied and vacant unit segments on P_{set}, we can find $Fr_o(e, e_f)$ and $Fr_v(e, e_f)$, respectively.

Given a unit segment $u = (r_i, \ldots, r_j)$ and corresponding path $P' = (e_s, e_{s+1}, \ldots, e_t)$, the road transition frequency of u is

$$Fr_o(u) = \frac{\sum_{k=s}^{t-1} Fr_o(e_k, e_{k+1})}{t - s + 1} \tag{13}$$

$$Fr_v(u) = \frac{\sum_{k=s}^{t-1} Fr_v(e_k, e_{k+1})}{t - s + 1}. \tag{14}$$

6) Taxi Cheating Frequency: Based on statistical analysis, we observe that a small amount of taxis have a high frequency of taking unmetered occupied trips. In order to capture these "recidivists," we obtain a taxi cheating frequency for each taxi as a statistical feature. Formally, cheating frequency is defined as

$$Fr(tid) = \frac{\mu_o}{\mu_v} \tag{15}$$

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

After extracting these features from the fragmented unmetered unit segments, we perform several post-processing tasks to calibrate the feature table. Since the spatial–temporal features (except time) are directly calculated from the data set, we normalize them to 0-1 range. Note that these features are flattened into vectors based on time, therefore the normalization is implemented within each column of each feature matrix. In the experiments, we find that taxi cheating frequency is not effective alone but achieves better performance as a weight combined with spatial–temporal features. Hence, we transform each spatial–temporal feature f in terms of

$$f = \Big\langle \underbrace{0, \ldots, 0}_{h}, f \times Fr(tid), \underbrace{0, \ldots, 0}_{\mu_t - h - 1} \Big\rangle. \tag{16}$$

Meanwhile, we calculate road transition frequency in order to capture the contiguity of the consecutive tracing records along trajectories. Therefore, this feature is used in maximum fraud trajectory construction, which will be described in Section VI-B.

Algorithm 2 Occupancy Detection

```
Require: trajectory $l$
Ensure: the occupancy status $r[o]$ for each tracing record $r \in l$
extract spatial-temporal features and statistical features from $l$ according to Section V-B
for each tracing point $r \in l$ do
    for each feature of $l$ do
        predict $r[o]$ via SGD using only this feature
    end for
    integrate the likelihoods of $r[o]$ based on Equation (18)
end for
return the resulting integrated occupancy status $u(o)$ for each tracing record $r \in l$
```

VI. Anomaly Detection

In this section, we introduce the anomaly detection phase, including occupancy detection and maximum fraud trajectory construction.

A. Occupancy Detection

Based on the features generated in Section V-B, we build an integrated predictor as shown in Fig. 5 to detect the occupancy status of taxis. More specifically, we first use stochastic gradient descent (SGD) on each feature to make a preliminary prediction of the occupancy status, and then integrate the likelihoods of each prediction result. More specifically, given a threshold λ and an instance u, the occupancy status of u is defined as

$$u(o) = \begin{cases} 1, & \frac{1}{n}\sum_{k=1}^{n} \mathscr{L}_i \geq \lambda \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where $\mathscr{L}_i$ is the prediction with the ith feature, and n is the number of utilized features. The pseudocode of the integrated predictor is shown in Algorithm 2.

In order to find occupied segments from the fragmented trajectories, there are two issues that need to be solved.

First, as introduced in Section I, the occupancy information contained in taxi trajectory data set is not precise; therefore, we need to find a way to obtain the ground truth. In this article, we use the data from "reliable" taxis as the ground truth. More specifically, given a taxi ID tid and a set of segmented metered trajectories L as described in Section IV-A, the occupancy reliability of taxi tid is defined as

$$R(tid) = \frac{\mu_{\text{occupy}}}{\mu_{\text{all}}} \tag{18}$$

where μ_{all} is the number of tracing records $r_i \in L$ where $r_i[id] = tid$, and μ_{occupy} is the number of $r_i \in L$ where $r_i[id] = tid$ and $r_i[o] = 1$. In this article, only taxis with reliabilities more than 90% are chosen.

Second, according to our observation, there are less than 5% of the unmetered unit segments are occupied among all the unmetered unit segments. With such skewed data distribution, regular classification models often fail to give a satisfactory result. To conquer the problem, we use the following training process.

1) The training set is divided into vacant and occupied feature sets with statistical features separately extracted.
2) The above two feature sets are partitioned into n_p groups. The size of each vacant group is 19 times larger than the size of each occupied group due to there are only 5% of occupied unit segments.
3) In order to balance the sampling size, each vacant group is equally split into 19 piles, thus resulting in n_p occupied piles and $19 \times n_p$ vacant piles.
4) During each training process, one occupied pile and one vacant pile are randomly picked and joined as a temporal training set.
5) The cross-validation is terminated when all the occupied feature sets have been used at least once.

B. Maximum Fraud Trajectory Construction

In Section VI-A, the detected occupied unit segments are discrete, thus it is necessary to implement a post-processing method that constructs a connected trajectory. The details of the maximum fraud trajectory construction algorithm is shown in Algorithm 3.

The algorithm heuristically searches the previous (line 14) and following (line 21) unit segments of each detected occupied unit segment $u \in U_t$. If the unit segment is initially detected as vacant, we use road transition frequency to infer the suspicion of it being occupied (lines 15 and 22). Formally, the suspicion of unit segment u being occupied is

$$S(u) = (1 + Fr_o(u) - Fr_v(u)) \times \mathscr{L}(u) \tag{19}$$

where $\mathscr{L}(u)$ is the likelihood of occupancy resulting from the model as described in Section VI-A. If the suspicion of a unit segment exceeds the threshold ϵ, it is inferred as occupied (lines 16 and 23) and the heuristic search continues. The heuristic search terminates when the next unit segment is still inferred as vacant or has already been detected as occupied (lines 18 and 25). After all the occupied unit segments $u \in U_t$ are checked, the consecutive unit segments that are occupied (lines 31 and 32) are connected. Finally, if the resulting trajectory is longer than γ, it concludes a maximum fraud trajectory (lines 33 and 34).

Algorithm 3 Maximum Fraud Trajectory Construction

```
Require: the sequence of unit segments $U = (u_1, \ldots, u_n)$;
    the set of occupied unit segments $U_t = \{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$
    where $1 \leq i_1 < i_2 < \cdots < i_k \leq n$; constant $\epsilon$ and $\gamma$
Ensure: the set of occupied trajectories $L_t$
 1: $L_t \leftarrow \emptyset$
 2: $U_o \leftarrow \{\{u_i\} - U_t{:}u_i \in U\}$
 3: for $j \leftarrow 1$ to $k$ do
 4:     if $j > 1$ then
 5:         $i_s \leftarrow i_{j-1} + 1$
 6:     else
 7:         $i_s \leftarrow 1$
 8:     end if
 9:     if $j < n$ then
10:         $i_t \leftarrow i_{j+1} - 1$
11:     else
12:         $i_t \leftarrow n$
13:     end if
14:     for $q \leftarrow i_j - 1$ to $i_s$ do
15:         if $U_o(q) \neq \emptyset$ and $S(u_q) \geq \epsilon$ then
16:             $U_o(q) \leftarrow \emptyset$
17:         else
18:             break
19:         end if
20:     end for
21:     for $q \leftarrow i_j + 1$ to $i_t$ do
22:         if $U_o(q) \neq \emptyset$ and $S(u_q) \geq \epsilon$ then
23:             $U_o(q) \leftarrow \emptyset$
24:         else
25:             break
26:         end if
27:     end for
28: end for
29: $l \leftarrow \emptyset$
30: for $j \leftarrow 1$ to $n$ do
31:     if $U_o(q) = \emptyset$ then
32:         $l \leftarrow l \cup \{u_j\}$
33:     else if $|l| \geq \gamma$ then
34:         $L_t \leftarrow L_t \cup \{l\}$
35:         $l \leftarrow \emptyset$
36:     end if
37: end for
38: if $|l| \geq \gamma$ then
39:     $L_t \leftarrow L_t \cup \{l\}$
40: end if
41: return $L_t$
```

VII. Evaluation

In this section, we will evaluate our method on both real-world and synthetic trajectory data sets.

A. Data Sets

1) Real-World Trajectories: The real-world taxi trajectories are provided by the transportation department of a large city in China. The data set contains four million taximeter records and 154 million taxi tracking records of 15 231 taxis [27]. Based

Algorithm 4 Generation of Synthetic Occupied Trajectory

```
Require: trajectory length $k$; random road segment $e^*$
Ensure: the synthetic trajectory $l = (e^*, e_2, \dots, e_k)$
while true do
    find a random path $P = (e_1, e_2, \dots, e_n)$, such that $|P| = n = k$ and $e_1 = e^*$
    if $P$ is the shortest path of $e_1$ and $e_n$ then
        $l \leftarrow \emptyset$
        for $e_i \in P$ do
            randomly select tracing record $r_i$ where $r_i[p] \in e_i$
            $r_i[o] \leftarrow 1$
            $l \leftarrow l \cup (r_i)$
        end for
        return $l$
    end if
end while
```

Algorithm 5 Generation of Synthetic Vacant Trajectory

```
Require: trajectory length $k$; random road segment $e^*$
Ensure: the synthetic trajectory $l = (r_1, r_2, \dots, r_k)$
$l \leftarrow \emptyset$
while $|l| < k$ do
    randomly select $e_i$ where $e_i$ and $e_{i-1}$ are connected
    randomly select tracing record $r_i$ where $r_i[p] \in e_i$
    $r_i[o] \leftarrow 0$
    $l \leftarrow l \cup (r_i)$
end while
return $l$
```

on our observation, there are 69% unmetered tracking records, and 5% of them are occupied. As described in Section VI, trajectories belong to the taxis with reliabilities greater than 90% are considered as ground truth.

The road network is also provided by the same transportation department and consists of 36 451 road segments with 25 613 road intersections.

2) Synthetic Trajectories: The real-world trajectories may not reflect all the possible variance of driving behaviors. Hence, in order to evaluate our system in all possible situations, we introduce a set of fine-grained synthetic trajectories for the evaluation. According to the observations in Section I, a fine-grained synthetic trajectory should follow the rules below.

1) A synthetic occupied trajectory should be moving in a local-optimal path, e.g., shortest path.
2) A synthetic vacant trajectory should be moving in a random path, as if the driver is hunting for passengers.

For a synthetic occupied trajectory, we use Algorithm 4 to synthesize the generated tracing records. In Algorithm 4, we first generate a random road segment as the starting point e^*, and then try to find all possible shortest paths starting with e^* within k hops of road segments. A shortest path fulfils the above requirements of synthetic rules. For a synthetic vacant trajectory, we populate tracing records in a random way, as shown in Algorithm 5. The random generation of tracing points fulfils the above requirements of synthetic rules.

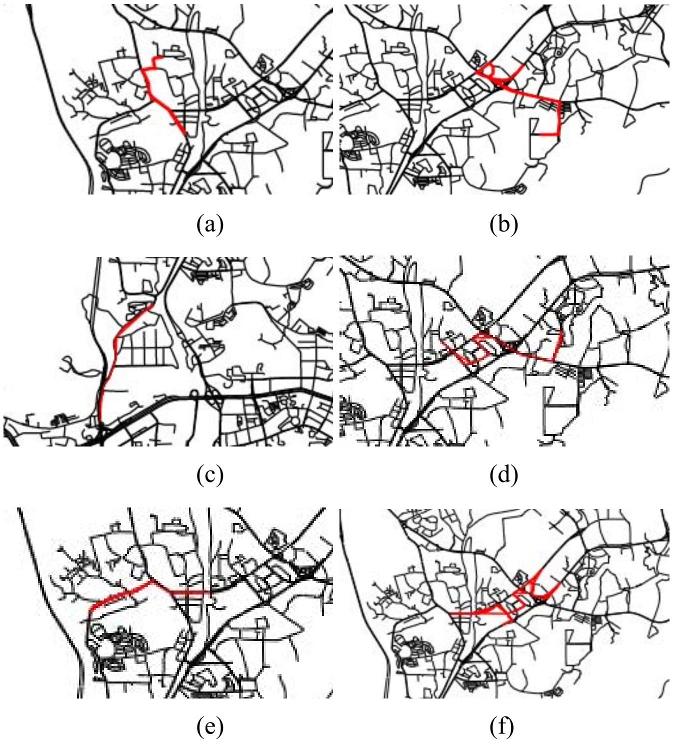

(a) (b) (c) (d) (e) (f)

Fig. 6. Examples of synthetic trajectories (in red color). (a), (c), and (e) Synthetic occupied trajectory. (b), (d), and (f) Synthetic vacant trajectory.

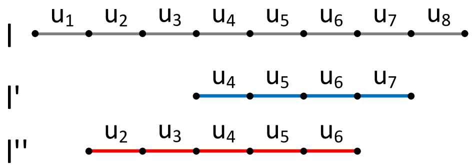

l
u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8
l'
u_4 u_5 u_6 u_7
l''
u_2 u_3 u_4 u_5 u_6

Fig. 7. Example of the recall defined in this article.

Based on the above synthetic rules, it is possible to evaluate the performance of synthesis via arc-chord ratio and curvature as introduced in Section V-B. Generally speaking, a vacant trajectory should have a higher arc-chord ratio and a higher curvature. Some examples of synthetic trajectories are shown in Fig. 6. It is obvious that the vacant trajectories on the right have a higher value in terms of both arc-chord ratio and curvature than the occupied trajectories on the left. In fact, the average arc-chord ratio and curvature of those synthetic occupied trajectories used in the experiments of this article are 7.5 and 46, respectively; while the average arc-chord ratio and curvature of those synthetic vacant trajectories used in the experiments of this article are 67 and 97, respectively.

B. Evaluation Metrics

In this article, the occupancy detection model is evaluated through *precision*, denoted as

$$\text{precision} = \frac{\mu_p^*}{\mu_p} \tag{20}$$

where μ_p and μ_p^* are the total number and correctly detected number of occupied unit segments, respectively.

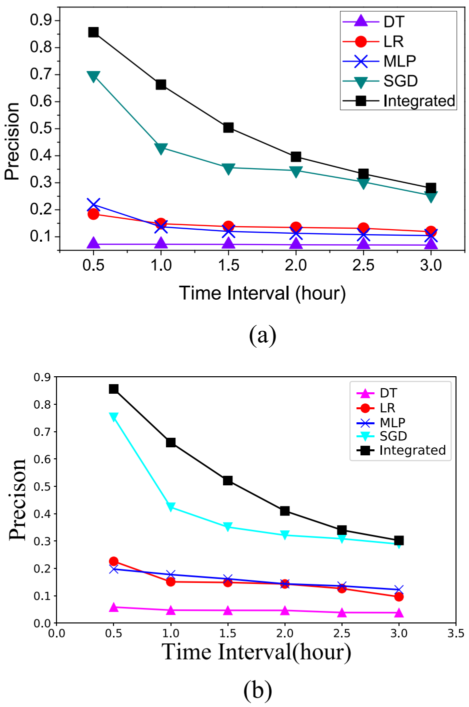


Fig. 8. Precision of occupancy detection under different Δ_t. (a) Real-world trajectories. (b) Synthetic trajectories.

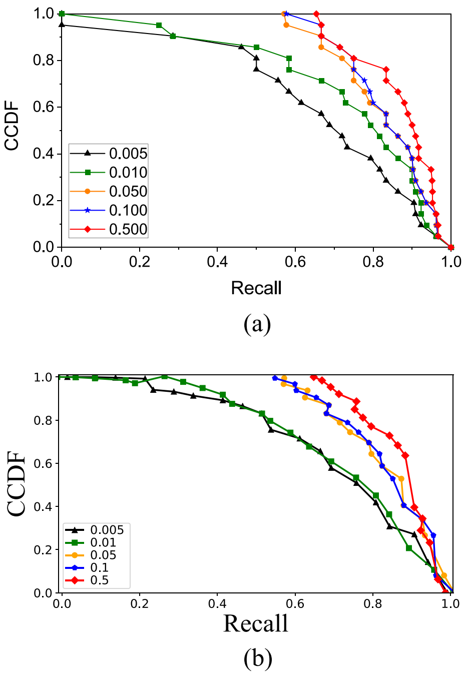


Fig. 9. CCDF of the recall of maximum fraud trajectory construction under different γ. (a) Real-world trajectories. (b) Synthetic trajectories.

Meanwhile, we evaluate the effectiveness of the maximum fraud trajectory construction algorithm as introduced in Section VI-B using *recall*. Formally, given a fraudulent trajectory $l' \subset l$ and a corresponding constructed trajectory l'', the recall of l'' is

$$\text{recall} = \frac{|l' \bigcap l''|}{|l'|} \qquad (21)$$

where $|l' \bigcap l''|$ is the number of unit segments in the intersection subtrajectory of l' and l''. For example, in Fig. 7, the trajectory l consists of eight unit segments $u_1, u_2, \ldots, u_8$, where u_4, u_5, u_6, u_7 is the real fraudulent trajectory l' according to the taximeter record. Suppose the maximum fraud trajectory l'' constructed by our algorithm consists of u_2, u_3, u_4, u_5, u_6, than the recall of l'' is $3/4 = 0.75$.

C. Experiment Results

1) Precision of Occupancy Detection: In the experiments, we use 30% of the trajectories as the training set, and the rest as the test set for both the synthetic and real-world data sets. The training set is partitioned with $n_p = 10$ for cross-validation. As explained in Section I, existing methods are not suitable for the detection of unmetered taxi trips; therefore, we compare the occupancy detection model with commonly used fraud detection models, including decision tree (DT), SGD, logistic regression (LR), and multilayer perceptron (MLP). The default values of each parameter described in Sections V and VI are: $w = 1$, $\lambda = 0.5$, and $\gamma = 0.5$.

In order to show the effectiveness of FraudTrip, we conduct six groups of experiments using our integrated predictor and each baseline prediction model by changing Δ_t from 30 min to 3 h. The results are shown in Fig. 8. It is clear that the integrated predictor has a much higher performance than the baselines under different Δ_t on both synthetic and real-world trajectory data sets. In particular, the precision of the integrated predictor is up to 85% whereas the baselines are less than 30%, when Δ_t is 30 min. Moreover, when Δ_t enlarges, precision decreases for all models, which indicates that time interval should be set as precise as possible, and it confirms the importance of time as a feature as introduced in Section V-B.

2) Recall of Maximum Fraud Trajectory Construction: The effectiveness of the maximum fraud trajectory construction algorithm as introduced in Section VI-B is evaluated by conducting five groups of experiments and changing γ from 0.005 to 0.5. The default values of each parameter described in Sections V and VI are: $w = 1$, $\lambda = 0.5$, and $\Delta_t = 30$ min.

We use complementary cumulative probability function (CCDF) to evaluate the coverage ratio of the fraudulent trajectories. The results are shown in Fig. 9, where the trajectory construction algorithm has the best performance on both synthetic trajectory data sets after γ reaches 0.5, and CCDF stays unchanged. Moreover, the results show that 87% of the synthetic trajectories and 81% of the real-world trajectories that are produced by the maximum fraud trajectory construction algorithm achieve over 75% coverage ratio, thus it indicates that the algorithm is effective [31].

3) Discussion of Parameter Configuration: The above experiments have discussed the configurations of Δ_t and γ for occupancy detection and maximum fraud trajectory construction, respectively. In this section, we will first discuss the impact of unit segment length w as introduced in Section V. The integration threshold λ as introduced in Section VI is not

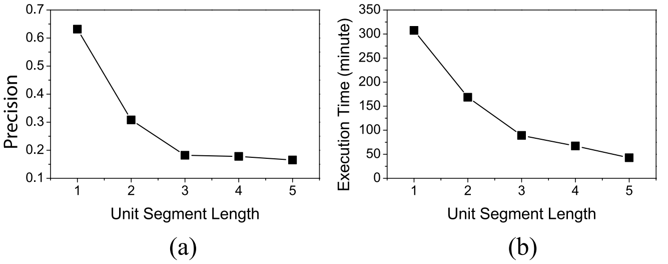


Fig. 10. Impact of unit segment length w on occupancy detection. (a) Precision. (b) Execution time.

discussed because it is obvious that a higher λ will result in higher precision but lower recall, and *vice versa*.

For the unit segment length w, we conduct a group of experiments with our integrated predictor and trajectory construction algorithm by increasing the values of w. Fig. 10(a) shows the impact of unit segment length on the precision of occupancy detection. It is clear that the precision of occupancy detection drops dramatically when the trajectories are fragmented into longer unit segments. Meanwhile, as shown in Fig. 10(b), the decline on the execution time of the entire fraud detection task is almost linear as the unit segment length increases. Therefore, there is a tradeoff between the effectiveness and efficiency upon w. In this article, in order to guarantee the effectiveness of FraudTrip, we set $w = 1$.

4) Analysis of Feature Performance: In order to interpolate the performance of each feature as introduced in Section V, we conduct experiments through FraudTrip by utilizing each spatial–temporal feature individually and each spatial–temporal feature weighted by taxi cheating frequency. Fig. 11 shows the results.

Consistent with the analysis in Section V-B, without considering taxi cheating frequency, the performance of spatial–temporal features is not satisfying, and the precision of occupancy detection rises significantly if we multiply taxi cheating frequency with each spatial–temporal feature. However, it is not effective to use taxi cheating frequency as an independent feature. Finally, the integrated predictor proposed in Section VI-A achieves the best performance by taking all the features into consideration.

D. Case Study

Fig. 12 illustrates six examples of the fraud trajectories detected by FraudTrip. In Fig. 12(a), the taxi is driving from a port of entry to a large park all along the large avenue. In Fig. 12(c), the taxi is driving from a high-speed railway station to a residential community. In Fig. 12(d), the taxi is driving from another port of entry to a large office complex. In Fig. 12(e), the taxi is driving from a crowded residential area to the central business district. In Fig. 12(f), the taxi is driving from a university to a down-town gymnasium.

These trajectories are highly suspicious after our investigation. First, the departures of these trajectories are traffic terminals. There are often people waiting to hire taxis at traffic terminals, and most of the taxis will successfully get a passenger after leaving the traffic terminal. Hence, it is very likely for a taxi to departure as occupied. Second, the arriving areas of these trajectories are often CBD or residential areas. Given the above information, it is unlikely for a taxi to leave the traffic terminal and try to find another passenger long away from a high-demanded area. Third, most of these trajectories are traveling through large avenues, which makes it faster for taxis to deliver passengers. However, this driving behavior is not efficient if the taxi is hunting for passengers.

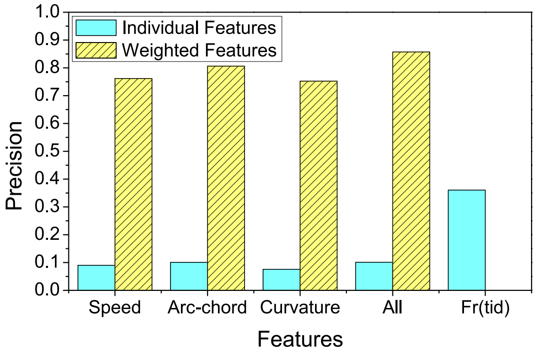


Fig. 11. Performance of different features in occupancy detection.

For comparison, we have also discovered some believable cases and one example is shown in Fig. 12(b). In Fig. 12(b), the source and destination locations of the trip is quite close, but the driving distance is 80 times longer than the shortest path. Moreover, the driver had made several hard u-turns during the trip. These behaviors are typical "hunting" behaviors and it is believable that this trajectory is indeed vacant.

In summary, the above case studies show that FraudTrip is both effective and efficient.

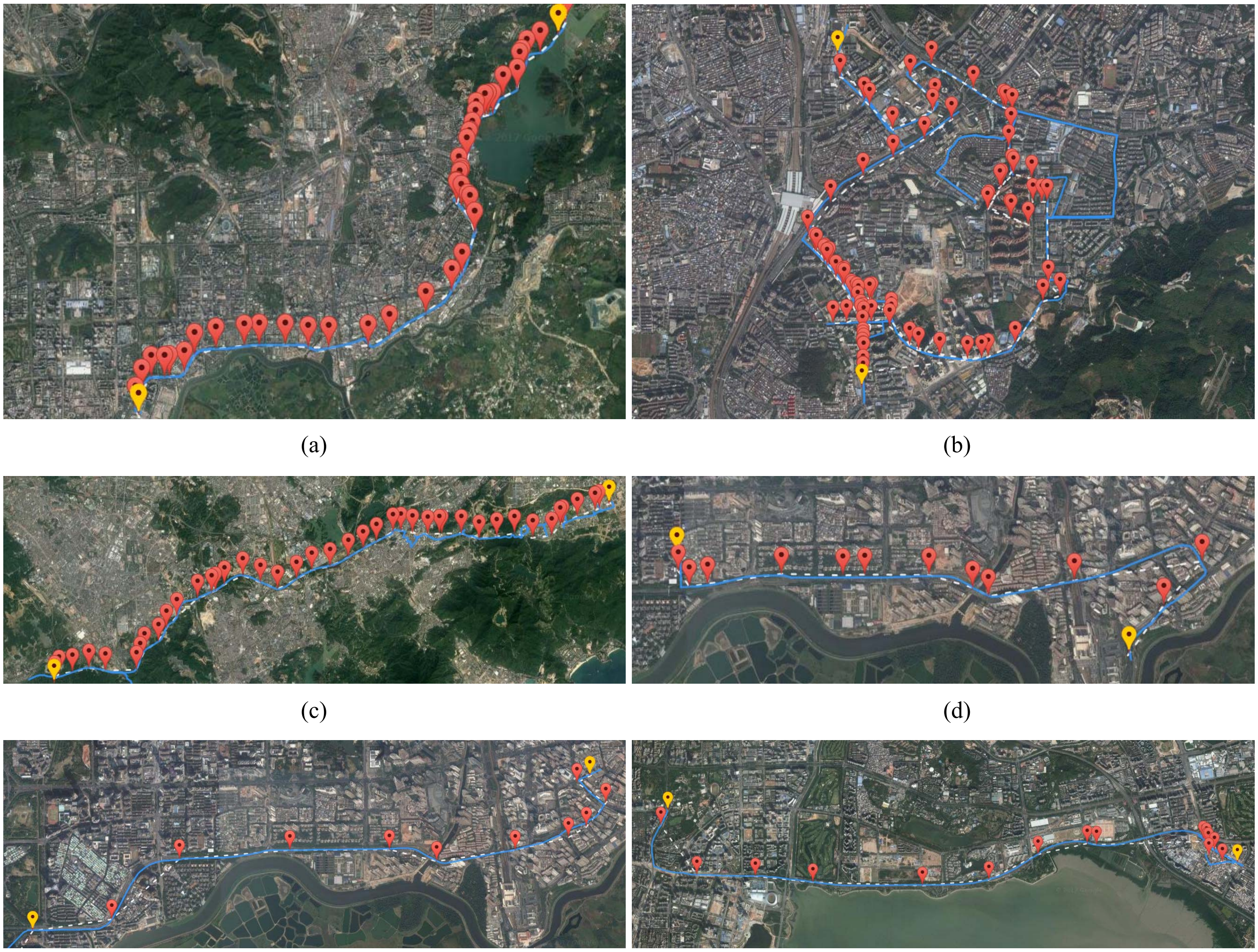
E. Discussion

As introduced in Sections V-B and VII-A2, arc-chord ratio and curvature are two significant features that are effective in predicting unmetered trajectories. Hence, in this section, we will discuss the performance of these two spectacular features.

1) Discussion of Arc-Chord Ratio: The arc-chord ratio introduced in Section V-B2 is defined as the measurement of taxi's driving distance upon the Euclidean distance between the source and destination locations. If arc-chord rations are large, we could suspect that the taxi may have taken the customer on a detour. This is because under normal situations, the taxi should be driving as fast as possible so that the driver would have a chance to pick up the next passenger. In this case, driving distance should be close to the Euclidean distance, and result in a small arc-chord ratio. To clearly show the effectiveness of the arc-chord ratio, Fig. 13 illustrates the CDF of arc-chord ratio of occupied and vacant taxis on both the real-world and synthetic trajectory data sets.

In Fig. 13, the CDF of the arc-chord ratios for occupied and vacant taxis differ a lot, thus it indicates that using the feature of arc-chord ratio could effectively interpolate the occupancy status.

2) Discussion of Curvature: Curvature is the representation of "straightness" of a trajectory, as introduced in Section V-B3. Typically, a taxi tends to drive in a straight-forward way to


(a)

(b)

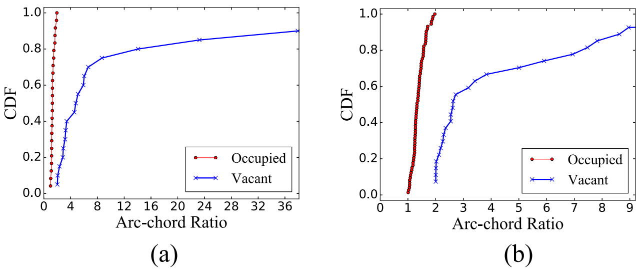
(c)

(d)

(e)

(f)

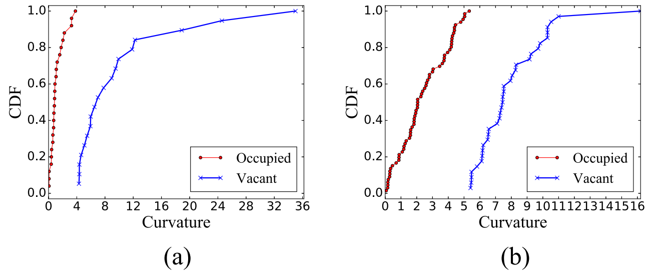
Fig. 12. Examples illustrating the effectiveness of FraudTrip. (a) Vacant case 1 (suspicious). (b) Vacant case 2 (believable). (c) Vacant case 3 (suspicious). (d) Vacant case 4 (Suspicious). (e) Vacant case 5 (Suspicious). (f) Vacant case 6 (Suspicious).



(a)

(b)

Fig. 13. CDF of the arc-chord ratios of occupied and vacant taxis. (a) Real-world trajectories. (b) Synthetic trajectories.



(a)

(b)

Fig. 14. CDF of the curvatures of occupied and vacant taxis. (a) Real-world trajectories. (b) Synthetic trajectories.

the destination so that the passenger could be delivered efficiently. Hence, it is intuitive for a trip being occupied if its curvature is small. This assumption is verified through the CDF of the curvatures of occupied and vacant taxis as shown in Fig. 14.

Similar to the discussion of arc-chord ratio, in Fig. 14, the CDF of the curvatures for occupied and vacant taxis differ significantly, thus it indicates that using the feature of curvature could effectively interpolate the occupancy status.

VIII. Conclusion

In this article, we considered *unmetered taxi trip* in real-world scenarios, which describes the taxi trip that has been recorded as vacant but has similar driving behaviors to regular metered trips. We design a novel system for the detection of unmetered taxi trips, called "FraudTrip," which consists of a learning model which predicts the occupancy status of taxis, and a heuristic algorithm which constructs anomalous unmetered trajectories. We demonstrate the effectiveness and efficiency of FraudTrip on both the synthetic and real-world taxi trajectory data sets. In this article, we have not considered the characteristics of the road networks and the influence of POIs, which will be studied in our future works.

REFERENCES

[1] X. Zhou, Y. Ding, F. Peng, Q. Luo, and L. M. Ni, "Detecting unmetered taxi rides from trajectory data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Boston, MA, USA, 2017, pp. 530–535.

[2] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "iBAT: Detecting anomalous taxi trajectories from GPS traces," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 99–108.

[3] S. Liu, L. M. Ni, and R. Krishnan, "Fraud detection from taxis' driving behaviors," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 464–472, Jan. 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, 2016, Art. no. e0171876.

[5] *Increasing the Taxi Fare Is a Dead End*. Accessed: Sep. 19, 2018. [Online]. Available: https://hk.lifestyle.appledaily.com/nextplus/magazine/article/20180919/2_624589_0/

[6] *146 Taxi Drivers Being Caught by Police in Guangzhou for Illegally Refusing or Arbitrarily Pricing Passengers*. Accessed: Feb. 6, 2017. [Online]. Available: http://www.gzcankao.com/news/wx/detail?newsi=39885

[7] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in *Proc. IEEE 24th Int. Conf. Data Eng. (ICDE)*, Cancun, Mexico, 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 *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discover. Data Min.*, 2009, pp. 159–168.

[9] Y. Ge, H. Xiong, Z.-H. Zhou, H. Ozdemir, J. Yu, and K. C. Lee, "Top-eye: Top-k evolving trajectory outlier detection," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manag.*, 2010, pp. 1733–1736.

[10] 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 *Proc. Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Serv.*, 2011, pp. 63–74.

[11] Z. Lv, J. Xu, P. Zhao, G. Liu, L. Zhao, and X. Zhou, "Outlier trajectory detection: A trajectory analytics based approach," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2017, pp. 231–246.

[12] V. Patil, P. Singh, S. Parikh, and P. K. Atrey, "GeoSClean: Secure cleaning of GPS trajectory data using anomaly detection," in *Proc. IEEE Conf. Multimedia Inf. Process. Retrieval (MIPR)*, Miami, FL, USA, 2018, pp. 166–169.

[13] J. Zhao *et al.*, "Unsupervised traffic anomaly detection using trajectories," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR Workshops)*, 2019, pp. 133–140.

[14] L. Lu, H. Cheng, S. Xiong, P. Duan, and Y. Xiao, "Distributed anomaly detection algorithm for spatio-temporal trajectories of vehicles," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl. Int. Conf. Ubiquitous Comput. Commun. (ISPA/IUCC)*, Guangzhou, China, 2017, pp. 590–598.

[15] H. Ergezer and K. Leblebicioğlu, "Anomaly detection in trajectories," in *Proc. IEEE 24th Signal Process. Commun. Appl. Conf. (SIU)*, Zonguldak, Turkey, 2016, pp. 1561–1564.

[16] J. Wang *et al.*, "Anomalous trajectory detection and classification based on difference and intersection set distance," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2487–2500, Mar. 2020.

[17] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1544–1554, Nov. 2008.

[18] J. Smith, I. Nouretdinov, R. Craddock, C. Offer, and A. Gammerman, "Anomaly detection of trajectories with kernel density estimation by conformal prediction," in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innovat.*, 2014, pp. 271–280.

[19] P.-R. Lei, "A framework for anomaly detection in maritime trajectory behavior," *Knowl. Inf. Syst.*, vol. 47, no. 1, pp. 189–214, 2016.

[20] D. Kumar, J. C. Bezdek, S. Rajasegarar, C. Leckie, and M. Palaniswami, "A visual-numeric approach to clustering and anomaly detection for trajectory data," *Vis. Comput.*, vol. 33, no. 3, pp. 265–281, 2017.

[21] W. Yang, Y. Gao, and L. Cao, "TRASMIL: A local anomaly detection framework based on trajectory segmentation and multi-instance learning," *Comput. Vis. Image Understand.*, vol. 117, no. 10, pp. 1273–1286, 2013.

[22] L. Sun, D. Zhang, C. Chen, P. S. Castro, S. Li, and Z. Wang, "Real time anomalous trajectory detection and analysis," *Mobile Netw. Appl.*, vol. 18, no. 3, pp. 341–356, 2013.

[23] N. Suzuki, K. Hirasawa, K. Tanaka, Y. Kobayashi, Y. Sato, and Y. Fujino, "Learning motion patterns and anomaly detection by human trajectory analysis," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Montreal, QC, Canada, 2007, pp. 498–503.

[24] X. Li, Z. Li, J. Han, and J.-G. Lee, "Temporal outlier detection in vehicle traffic data," in *Proc. IEEE 25th Int. Conf. Data Eng. (ICDE'09)*, Shanghai, China, 2009, pp. 1319–1322.

[25] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Time-dependent popular routes based trajectory outlier detection," in *Proc. Int. Conf. Web Inf. Syst. Eng.*, 2015, pp. 16–30.

[26] Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou, "A taxi driving fraud detection system," in *Proc. IEEE 11th Int. Conf. Data Min. (ICDM)*, Vancouver, BC, Canada, 2011, pp. 181–190.

[27] Y. Ding, S. Liu, J. Pu, and L. M. Ni, "HUNTS: A trajectory recommendation system for effective and efficient hunting of taxi passengers," in *Proc. IEEE 14th Int. Conf. Mobile Data Manag. (MDM)*, vol. 1. Milan, Italy, 2013, pp. 107–116.

[28] X. Zhou, Y. Ding, H. Tan, Q. Luo, and L. M. Ni, "HIMM: An HMM-based interactive map-matching system," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2017, pp. 3–18.

[29] C. Chen, K. Li, S. G. Teo, X. Zou, K. Li, and Z. Zeng, "Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks," *ACM Trans. Knowl. Discover. Data (TKDD)*, vol. 14, no. 4, pp. 1–23, 2020.

[30] C. Chen *et al.*, "Gated residual recurrent graph neural networks for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 485–492.

[31] D. Zwillinger and S. Kokoska, *CRC Standard Probability and Statistics Tables and Formulae*, Boca Raton, FL, USA: CRC Press, 1999.

Ye Ding (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2014, under the supervision of Prof. L. M. Ni.

He is currently an Associate Professor with the School of Cyberspace Security, Dongguan University of Technology, Dongguan, China. His research interests are spatial–temporal data analytics, big data, and machine learning.

Wenyi Zhang (Member, IEEE) is currently pursuing the postgraduation degree with the Dongguan University of Technology, Dongguan, China, under the supervision of Prof. Y. Ding.

His research interests include transportation and trajectory data analytics, spatial–temporal data analytics, and machine learning.

Xibo Zhou (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2018, under the supervision of Prof. L. M. Ni and Prof. Q. Luo.

His research interests include spatial–temporal data mining and distributed systems.

Qing Liao (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2016, under the supervision of Prof. Q. Zhang.

She is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China. Her research interests include big data analytics and artificial intelligence.

Qiong Luo (Member, IEEE) received the B.S. and M.S. degrees in computer sciences from Beijing (Peking) University, Beijing, China, in 1992 and 1997, respectively, and the Ph.D. degree in computer sciences from the University of Wisconsin–Madison, Madison, WI, USA, in 2002.

She is an Associate Professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. Her research interests are in big data systems, parallel and distributed systems, and scientific computing. Current focus is on data management on modern hardware, GPU acceleration for data analytics, and database support for e-science.

Lionel M. Ni (Life Fellow, IEEE) received the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 1980.

He is the Provost of the Hong Kong University of Science and Technology, Hong Kong, where he is the Chair Professor with the Department of Computer Science and Engineering.

Dr. Ni has chaired over 30 professional conferences and has received eight awards for authoring outstanding papers. He serves on the editorial boards of *Communications of the ACM*, the IEEE TRANSACTIONS ON BIG DATA, and the *ACM Transactions on Sensor Networks*. He is a Fellow of the Hong Kong Academy of Engineering Science.