A Taxi Driving Fraud Detection System

Yong Ge[1], Hui Xiong[1,*], Chuanren Liu[1], Zhi-Hua Zhou[2]
[1] *Rutgers Business School, Rutgers University*
yongge@pegasus.rutgers.edu, {hxiong,chuanren.liu}@rutgers.edu
[2] *National Key Laboratory for Novel Software Technology, Nanjing University, China*
zhouzh@lamda.nju.edu.cn

***Abstract*—Advances in GPS tracking technology have enabled us to install GPS tracking devices in city taxis to collect a large amount of GPS traces under operational time constraints. These GPS traces provide unparallel opportunities for us to uncover taxi driving fraud activities. In this paper, we develop a taxi driving fraud detection system, which is able to systematically investigate taxi driving fraud. In this system, we first provide functions to find two aspects of evidences: travel route evidence and driving distance evidence. Furthermore, a third function is designed to combine the two aspects of evidences based on Dempster-Shafer theory. To implement the system, we first identify $interesting\ sites$ from a large amount of taxi GPS logs. Then, we propose a parameter-free method to mine the travel route evidences. Also, we introduce $routemark$ to represent a typical driving path from an interesting site to another one. Based on $routemark$, we exploit a generative statistical model to characterize the distribution of driving distance and identify the driving distance evidences. Finally, we evaluate the taxi driving fraud detection system with large scale real-world taxi GPS logs. In the experiments, we uncover some regularity of driving fraud activities and investigate the motivation of drivers to commit a driving fraud by analyzing the produced taxi fraud data.**

***Keywords*-Taxi Driving Fraud; Location Traces; Dempster-Shafer Theory**

I. Introduction

Taxi driving frauds are often committed by greedy taxi drivers who overcharge passengers by deliberately taking unnecessary detours. Nowadays, many taxi service complaints are related to taxi driving frauds [1], [2], [3]. Therefore, it becomes invaluable for improving taxi services by providing the information about taxi driving frauds. However, it is a challenging issue to detect driving fraud activities committed by experienced and cunning taxi drivers who know how to manipulate the driving routes to commit driving frauds without being disclosed by passengers.

A promising direction to solve this problem is to collect and analyze the GPS traces by taxi drivers. Indeed, GPS tracking devices have been installed on taxis of many cities and a large amount of GPS traces has been accumulated for the analysis [4], [5], [6] . These GPS traces provide unparallel opportunities for us to develop new ways to uncover taxi driving fraud activities. To that end, in this paper, we develop a taxi driving fraud detection system by exploiting a large amount of GPS traces by taxi drivers.

However, it is a non-trivial task for detecting taxi driving frauds from GPS traces by taxi drivers. There are some inherent complexities involved in taxi driving fraud activities. For instance, Figure 1 shows the taxi traces between two locations, which are the source node S and the destination node E respectively. These trajectories were produced by different taxi drivers who delivered customers from S to E. By carefully visual examination, we can observe that there may be various types of driving frauds committed by different taxi drivers. For example, some suspicious driving traces significantly deviate from the majority of trajectories from S to E , such as the *black one* in Figure 1. Also, some suspicious driving activities (trajectories), such as the *red one* in Figure 1, lead to abnormally longer driving distances compared to the majority of trajectories.

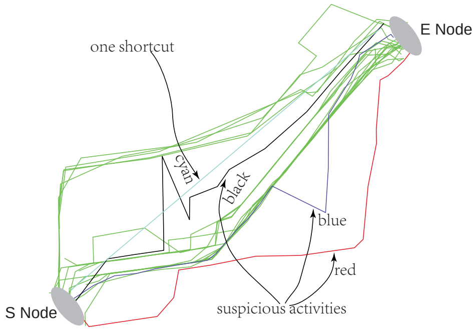


Figure 1. An Illustration of Taxi Driving Frauds.

Based on the above observations, a straightforward idea is to exploit traditional anomaly detection techniques for finding anomaly trajectories. For instance, traditional trajectory outlier detection approaches [7], [8] based on distance or density can be applied here to detect the anomaly trajectories. Also, model-based or clustering-based methods can be applied for exploiting the driving distance to detect the suspicious activities with abnormal driving distances. However, even with these anomaly detection techniques, we are still facing several essential challenges in order to develop a reliable taxi driving fraud detection system.

First, we need to deal with different sets of trajectories between different pairs of source and end nodes. These

[*] Contact Author.

different sets of trajectories are heterogeneous in nature and usually have different densities or distance distributions. It will be difficult to adjust parameters if we use traditional trajectory outlier detection approaches [7], [8]. In addition, some suspicious trajectories detected by density/distance-based methods may be just shortcuts instead of frauds, such as the *cyan one* in Figure 1.

Second, if we directly cluster or model driving distance samples for individual pair of source and end nodes, spatial information associated with each observation of driving distance will be missed. Also, there are usually multiple routes from source to end nodes, which leads to different driving distance. Subsequently, some suspicious activities, such as the *blue one* in Figure 1, which detours locally, may not be detected, since the corresponding driving distance may be still close to that of the majority trajectories.

Third, the use of most trajectory outlier detection methods can only detect certain types of taxi driving frauds. In fact, the unique characteristics of taxi driving fraud activities is that all drivers, who committed frauds, have the same motivation to overcharge customers. However, different driving habits and heterogeneous geo-context usually lead to diverse abnormal location traces than many other application scenarios, such as detecting the abnormal moving objects in the parking lot. A better way is to identify the key fraud characteristics of taxi driving fraud activities and use them together as the evidences to detect taxi driving frauds in a combined way.

Finally, in real-world applications, a practical challenge is that some drivers, who commit a driving fraud, may be truly unfamiliar with the local region or may use this as an excuse. Also, for some suspicious activities, taxi drivers may also argue that they changed driving routes due to the heavy traffic or car accidents. To effectively investigate these excuses, we need to further find out the pertinent evidences which can be leveraged in the taxi fraud detection system.

Specifically, in this paper, we develop a taxi driving fraud detection system which is equipped with three main functions to deal with the above challenges. To implement this system, we first identify *interesting sites* from the taxi GPS logs. These sites are frequently visited as pick-up or drop-off locations. All these $interesting\ sites$ can be source and end nodes. Between each pair of source and end nodes, we perform driving fraud detection. In detail, we provide two functions to discover two aspects of evidence: travel route evidence and driving distance evidence. To find travel route evidences, we propose a parameter-free method to compute the degree of frauds in terms of deviation, instead of using density or distance-based trajectory outlier detection techniques. To discover driving distance evidences, we first propose a $routemark$-based method to determine different routes from source to end nodes. A typical driving route from source to end nodes is represented with a sequence of $routemarks$. Then, we introduce generative statistical modeling to model the distribution of driving distance by incorporating the spatial information. The driving distance evidence is defined through the probability inference.

Given the discovered two aspects of evidences, we provide the third function to combine these two evidences. Specifically, we explore the Dempster-Shafer evidence theory to combine both evidences. The Dempster-Shafer evidence theory is well suitable for this type of problem for two reasons. First, it reflects uncertainty or a lack of complete information. Second, the Dempster's rule for combination gives a convenient numerical procedure for fusing together multiple evidences. After detecting all the driving fraud activities using the combined evidences, the taxi drivers who frequently commit the suspicious driving fraud activities can be identified. Moreover, we also provide some additional functions to deal with the possible excuses from fraudulent taxi drivers by checking their relevant historical traces.

In summary, the major contributions are as follows.

- First, we develop a taxi driving fraud detection system, which are equipped with several effective functions to identify taxi driving frauds. To the best of our knowledge, this is the first work on taxi driving fraud detection by effectively considering multiple evidences generated from Taxi GPS traces.
- Second, we encode the trajectories by using $symbol$ and oversampling, and disclose the driving route evidences via the coding cost. Also, we introduce the notion of $routemark$ to represent a typical driving path from an interesting site to another one, and then model the driving distance by generative statistical modeling.
- Third, we provide a case study by exploiting real-world GPS logs of around 500 taxi drivers during the period of 30 days, and evaluate the developed system in an organized way.
- Finally, we uncover some interesting regularities of taxi driving fraud activities by analyzing all the detected driving fraud activities. Also, we investigate a natural motivation of taxi drivers to commit a driving fraud by analyzing the taxi driving fraud data.

II. Preliminaries

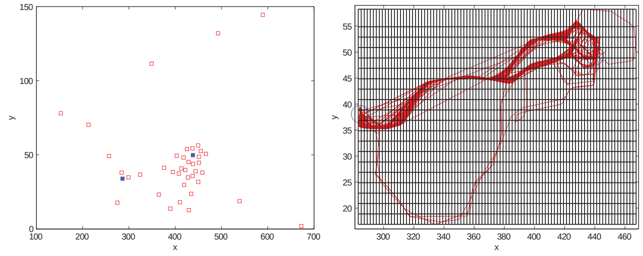
Here, we introduce some preliminaries about interesting site selection, symbol generation, and oversampling.

A. Interesting Site Selection

From the GPS traces by taxi drivers, we can obtain a large number of pick-up (source) and drop-off (end) points. Some source/end points are very close and correspond to the same specific area, namely $interesting\ site$, such as a supermarket or restaurant, because such $interesting\ sites$ are the places where people often depart from or arrive at in their daily life. In other words, there is the clustering effect of source/end points. To this end, we apply the clustering techniques on the source/end points and treat each cluster as one $interesting\ site$. In addition, some of the $interesting\ sites$ are not frequently-visited by taxis. So the reliable driving information, such as frequently-taken routes and the common driving distance/time from or to those $interesting\ sites$, cannot be obtained. Thus, we filter out some interesting sites not frequently visited for reliability.

B. Symbol Generation and Oversampling

With the selected $interesting\ sites$ as shown in Figure 2 (a), we can easily find the driving traces between $interesting\ sites$, which are produced by different drivers who deliver customers from one $interesting\ site$ to another one. For example, in Figure 2 (b), we pick two $interesting\ sites$ (blue shaded dots) from Figure 2 (a) and plot the trajectories from the source node (left) to the end node (right). Given these trajectories, source and end nodes, we discretize the continuous space into small grids as shown in Figure 2 (b). The grid size can be empirically specified by the user. This grid space will serve as a basis to help us discover travel route evidence and $routemark$.



(a) Interesting Sites

(b) Trajectories from Source to End Nodes

Figure 2. An Illustration

After this space discretization, we can represent each trajectory as an ordered sequence of grids, which contain the recorded points of the trajectory. Here, we denote each cell as a $symbol$ and all cells as an $alphabet$. This representation will help us to determine the travel route evidence as described in section III. However, one practical challenge is that the taxi GPS data is usually low-sampling-rate data. For example, the sampling rate of our real-world data is about 50 seconds. This low-sampling-rate will lead to non-detailed representation with $symbols$ for each trajectory, because a taxi may consecutively traverse multiple cells with no points being recorded. To deal with this challenge, we adopt oversampling technique, which is broadly used to increase resolution in the signal processing area, to insert pseudo recorded points. Here, we perform neighborhood-based oversampling by combing the direction information of the trajectory. Specifically, for each recorded point of the trajectory, such as p of T_1 in Figure 3, we insert one point in the neighboring grid(s) along the direction of the trajectory if there is no recorded points in the neighboring grid(s). The coordinates of the inserted point is specified as the center of the corresponding neighboring grid. Please note that we check 8 neighbors for each grid, such as the 8 grey cells around p in Figure 3, and select the neighboring ones along the direction of the trajectory. Eventually, this oversampling process allows us to obtain more fine representation with $symbols$ for trajectories.

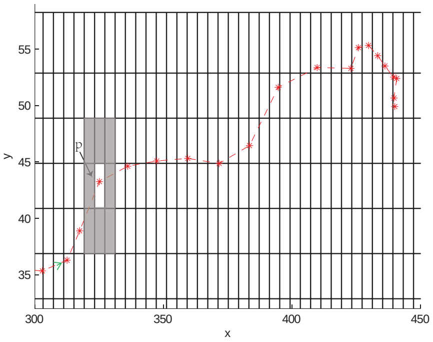


Figure 3. Trajectory T_1: Oversampling.

III. Travel Route Evidence

In this section, we introduce a parameter-free method to mine the travel route evidence.

With the extracted $symbols$, all driving activities (trajectories) from source node and end node can be further encoded by assigning a codeword to each $symbol$. From this perspective, a trajectory is considered as a driving fraud, if its coding cost is unusually high. According to coding theory [9], [10], the coding cost of a trajectory can be measured by the length of all involved codewords and the length of each codeword is usually decided as:

$$l_i = \left\lceil \log_2 \left(\frac{1}{p_i} \right) \right\rceil, \qquad (1)$$

where p_i is estimated probability of the extracted $symbol\ i$ and $\lceil x \rceil$ is the smallest integer greater than or equal to x.

So far, it seems that we can directly calculate the coding cost for each trajectory by adding the length of involved codewords for each trajectory. However, an unrealistic assumption that all $symbols$ are independent is used for this simple calculation. This unrealistic assumption results in unreliable estimation of coding cost. In fact, there is strong correlation between different $symbols$ due to the naturally spatial relationships among $symbols$. To address this, we propose to apply the independent component analysis (ICA) technique to the extracted $symbols$ in order to find independent components. Then, the trajectories are represented with the independent components instead of the original $symbols$. Finally, we define the travel route evidence as the coding cost with independent components.

In the following, we elaborate independent component analysis and the definition of travel route evidence.

A. Independent Component Analysis

Independent component analysis finds the independent components by maximizing the statistical independence of the estimated components. A broad definition of independence for ICA is maximization of non-Gaussianity. This maximization of non-Gaussianity naturally leads to the minimal coding cost, which is measured by entropy, because the entropy of Gaussian distribution is maximal.

To apply the ICA algorithm to GPS data, we treat trajectories as observations and all involved $symbols$ as attributes. Suppose there are M involved $symbols$ and N trajectories between one pair of source and end nodes. By ignoring the sequential order of $symbols$ in a trajectory, we can summarize the data in a $N \times M$ co-occurrence table, each element of which denotes how often one $symbol$ occurs in a trajectory. Before applying an ICA algorithm to the data, it is usually very useful to do two aspects of preprocessing: centering and whitening [11]. These preprocessing techniques make the problem of ICA estimation simpler and better conditioned. Formally, we denote by $\mathbf{x}$ the random vector, each variable of which corresponds to a involved $symbol$. Each variable (corresponding to a $symbol$) is a linear mixture of independent components. Note that each trajectory represented with the involved $symbols$ is one sample of the random vector $\mathbf{x}$. First, we center data $\mathbf{x}$ by subtracting its empirical mean $\mathbf{m}$ in order to make $\mathbf{x}$ a zero-mean variable. Second, after centering, we whiten the centered data $\mathbf{x_c}$ in order to de-correlate and normalize $\mathbf{x_c}$: $\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{x_c}$. $\mathbf{E}$ is the orthogonal matrix of eigenvectors of covariance matrix $E\{\mathbf{x_c}\mathbf{x_c}^T\}$ and $\mathbf{D}$ is the diagonal matrix of its eigenvalues, i.e., $\mathbf{D} = diag(d_1, \cdots, d_n)$. Also, $\mathbf{D}^{-1/2}$ is computed as $\mathbf{D}^{-1/2} = diag(d_1^{-1/2}, \cdots, d_n^{-1/2})$.

After these two preprocessing steps, we use the FastICA algorithm[12] to determine a weighting matrix $\mathbf{W}$ and independent components as $\mathbf{s} = \mathbf{W}^T\tilde{\mathbf{x}}$. The FastICA is based on a fixed-point iteration scheme for finding a maximum of the nongaussianity of $\mathbf{W}^T\tilde{\mathbf{x}}$, where the weight vectors are updated with the following rule:

$$\mathbf{W}^+ := E\{\tilde{\mathbf{x}}g(\mathbf{W}^T\tilde{\mathbf{x}})\} - E\{g'(\mathbf{W}^T\tilde{\mathbf{x}})\}\mathbf{W} \qquad (2)$$
$$\mathbf{W} = \mathbf{W}^+/||\mathbf{W}^+||$$

We use $tanh(x)$ for the non-linear contrast function $g(x)$. And $g'(x)$ is the derivation of $g(x)$ and $E\{\}$ is the expected value. $\mathbf{W}$ is updated until convergence.

The overall transformation of original data into the white space of independent components can be derived as:

$$\mathbf{s} = \mathbf{W}^T\mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T(\mathbf{x} - \mathbf{m}). \qquad (3)$$

Also, from this equation, we can easily get the inverse transformation and the weight of independent components.

B. Coding Cost

After applying the independent component analysis on the $symbols$, we represent each trajectory with independent components together with its weights, instead of $symbols$. Given the independence of components, the coding cost of each trajectory can be estimated by adding the weighted sum of coding cost of all the independent components involved in the trajectory. Since using variant distribution assumptions to approximate the probability of each independent component may lead to bias, we simply treat each independent component as a latent $symbol$ and estimate the probability of each latent $symbol$ via its relative frequency which corresponds to a popular interpretation of probability. Specifically, for each latent $symbol$, we add the corresponding weights across all trajectories and treat the sum as its frequency. The relative frequency can be obtained by normalizing over all latent $symbols$. Given the estimated probability, the coding cost of each latent $symbol$ is computed as Equation 1. Finally, we define the travel route evidence as the coding cost of each driving activity (trajectory).

IV. Driving Distance Evidence

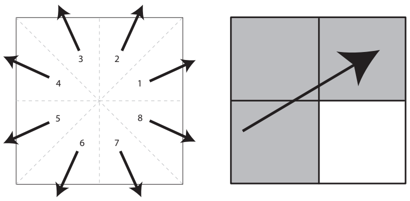
In this section, we introduce a $routemark$-based method to mine the driving distance evidence.

From a source node to an end node, there are usually several different routes. These different driving paths result in the mixed distribution of driving distances from the source node to the end node. To mine the anomaly in terms of driving distance, we first find frequently-used driving routes via the constructed $routemarks$. Then, we use generative statistical modeling to model the distribution of driving distances by treating the frequently-used driving routes as the prior information. Finally, the anomaly degree in terms of driving distance is defined as an inferred probability.

A. RouteMark Construction

In this paper, a $routmark$ from a source node to an end node is defined as a road segment, which is frequently passed by taxis driving from the source node to the end node. We discover $routemark$ from a source node to an end node based on the $symbol$ (cell) and oversampling introduced in subsection II-B. First, we summarize the direction information within each cell. Specifically, we partition each grid into 8 direction bins, as shown in Figure 4 (a), each bin with a angle range of $\pi/4$. The goal of this partition is to summarize the direction information within grids using the involved trajectories and represent it with a direction vector. Specifically, we represent the direction information of each grid with a direction vector as: $g = (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8)$, where p_i is the probability of moving along direction bin i within this grid. To compute p_i, we first count the frequency f_i of moving trajectories which have passed this grid and have the direction along the direction i, i.e. within direction bin i. For example, for the monitoring area as shown in Figure 4 (b), a vector is across three grids $< 1, 1 >$,$< 1, 2 >$, and $< 2, 1 >$ along direction 1. Therefore, the frequency of direction 1 will increase by one for all these three grids.

Then, $p_i = f_i/n$, where $n = \sum_{i=1}^{8} f_i$. Therefore, p_i is the probability of the moving towards the direction i within this grid. This transformed direction vector for each grid is more suitable for similarity measures than the original trajectories. Also, these kind of discrete direction bins eliminate the slight direction variance within each direction bin, and thus is better for direction summarization and comparison. In the meantime, the density of each grid can be computed by scanning each individual trajectory and increasing the density of all the grids by one where this trajectory passes.



(a) Direction Distretization

(b) Grid Counting

Figure 4. Data Transformation.

After obtaining the direction and density summary of each cell, we perform a join processing to construct $routemark$. Basically, a $routemark$ from a source node to an end node is a sequence of cells, which are spatially connected and have similar summarized direction and density. As shown in algorithm 5, the idea of this join processing is that we iteratively merge two cells, which are neighbors and most similar in terms of direction and density. Here, we use the cosine similarity measurement. As in subsection II-B, we check the 8 neighbors of each (pseudo) cell for the comparison of similiarty.

```
Input:
        $C$: all involved cells together with direction and
            density summary
        $\tau$: similarity threshold
Output:
        $M$: all routemarks.
Begin iteration
 1.     find the neighboring two cells with largest
            similarity $s$ and $s > \tau$
 2.     merge these two cells:
            a. average the direction and density summary
                of these two cells;
            b. treat these two cells as one merged pseudo cell;
 3.     update $C$ by replacing the two cells with the pseudo cell
Until no two cells are merged
$M$ is the set of all pseudo cells and remaining cells
```

Figure 5. RouteMark Construction

With the identified $routemarks$, we can represent each trajectory from a source node to an end node as a directed sequence of $routemark$s (road segments). To further discover the frequently-used routes from a source node to an end node, we simply count the support of each sequence of $routemark$ and maintain the sequences with high relative support. Here, the support count of one sequence of $routemark$ is the number of trajectories, which are represented by this sequence. And the relative support of one sequence of $routemark$ is the percentage of trajectories, which are represented as this sequence. Finally, each maintained directed sequence is considered as one route r from a source node to an end node.

B. Driving Distance Modeling

We use generative statistical modeling to model the distribution of driving distance from a source node and an end node. Specifically, we treat the frequently-used routes as prior and the driving distance is considered as observation.

Suppose we discover K mainly-used routes from a source node and an end node. We denote each route as r. Also, we have N observations for driving distance and denote each observation of driving distance as l. The set of observations of driving distance are conditionally independent given the prior. Then, we generatively model all observations of driving distance as: $p(L|R) = \prod_{i=1}^{N} p(l_i|r_i)$, where $p(l_i|r_i)$ is the conditional probability of having the observation l_i of driving distance given the driving route r_i, which is associated with the observation. L and R represent the set of all driving-distance observations and the set of all driving-route observations respectively. We assume that the driving distance distribution given prior $r_j (j = 1, \cdots, K)$ is characterized by Gaussian distribution parameter,

$$p(l|r_j) = \mathcal{N}(l|\mu_j, \sigma_j^2), \tag{4}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ denotes the probability density function of the Gaussian distribution with mean μ and variance σ^2. Thus, K pairs of parameters, i.e.,$[\mu_j, \sigma_j^2]$ $(j = 1, \cdots, K)$, need to be estimated for this generative statistical model. For simplicity, we assume $\sigma_j^2 = \sigma^2$ for $j = 1, \cdots, K$. We use the popular MLE method to estimate these parameters.

After estimating the parameters, we define the anomaly degree d of a new trajectory with length l from the given source to end nodes in terms of driving distance as:

$$d(l) = 1 - \sum_{j=1}^{K} p(l|r_j)p(r_j). \tag{5}$$

Here, $p(r_j)$ is the prior probability. Instead of estimating this prior from all the observations of driving distance, which does not take into account the spatial-context information, we leverage the $routemark$ to approximate this prior. Specifically, this prior is estimated as the ratio of length of all the $routemarks$, which belong to route r_j and are passed by the new trajectory l, to the total length of the new trajectory l. After this estimation, we apply the normalization to the estimations of prior distribution in order to hold the constraints, i.e., $\sum_{j=1}^{K} p(r_j) = 1$. In addition, $p(l|r_j)$ is calculated with the learned Gaussian parameters.

V. Dempster-Shafer Evidence

To combine the identified two evidences, we exploit the Dempster-Shafer Theory, which aims to combine evidences from different sources and yields a degree of belief that takes into account all the available evidences. This actually aligns very well with our goal that is to determine the taxi driving fraud by combining the two aspects of evidences.

The Dempster-Shafer Theory [13] is a mathematical theory of evidence which is considered to be a generalization of the Bayesian theory of subjective probability. Unlike Bayesian methods which often require a complete probabilistic model, the Dempster-Shafer framework does not specify priors and conditionals. The Dempster-Shafer theory is based on two ideas. The first idea is the notion of obtaining degrees of belief for one question based on subjective probabilities for a related question. The second one is the Dempster's rule for combining such degree of belief when they are based on independent items of evidence. Since we obtain two independent evidence, travel route evidence and driving distance evidence, we are interested in the latter part of the Dempster-Shafer theory, namely, Dempster's rule.

In the Dempster-Shafer theory, a frame of discernment (a universe of discourse) is a set of mutually exclusive and exhaustive possibilities denoted by Ω, which is similar to a state space in probability. Any hypothesis will refer to a subset of Ω. The set of all possible subsets of Ω, including itself and the null set $\varnothing$, is called a power set and denoted as 2^{Ω}. Thus, the power set consists of all possible hypotheses or so-called elements. Suppose we want to combine evidence for a hypothesis H, which is a member of the power set 2^{Ω}, and we have two independent sources of evidence m_1 and m_2, the Dempster's rule combines them in the following frame:

$$m_{1,2}(H) = \frac{\sum_{A,B\subseteq\Omega, A\cap B=H} m_1(A)m_2(B)}{\sum_{A,B\subseteq\Omega, A\cap B\neq\varnothing} m_1(A)m_2(B)}. \tag{6}$$

Here, A and B are supersets of H. They are not necessarily proper supersets; that is, they may be equal to H or to the frame of discernment Ω. $m_1(A)$ is the portion of belief assigned to A by m_1. $m_{1,2}(H)$ is the combined Dempster-Shafer probability for a hypothesis H.

To elaborate more about the Dempster-Shafer theory, we present the following example. For a suspected driving activity (trajectory) R from a source node to an end node, we are interested in determining if this suspected activity is a driving fraud or not. Thus we may form the frame of discernment, which consists of two possibilities, as $\Omega = \{T, \bar{T}\}$, where T means R is driving fraud, and $\bar{T}$ means it is not. For this Ω, we may form the following propositions (elements) which correspond to the power set of Ω together with the null set $\varnothing$:

Hypothesis $H = \{T\}$: R is driving fraud;
Hypothesis $\bar{H} = \{\bar{T}\}$: R is not driving fraud;
Hypothesis $U = \Omega$: R is either driving fraud or not.

Since we extract travel route evidence and driving distance evidence, we have two independent sources of evidence m_1 (travel route evidence) and m_2 (driving distance evidence). Suppose the probability of travel route evidence being trustworthy is α. If travel route evidence claims the driving activity R is a fraud, its basic uncertainty assignment will be:

$m_1(H) = \alpha$;
$m_1(\bar{H}) = 0$;
$m_1(U) = 1 - \alpha$.

If travel route evidence claims the trajectory R as a non-fraud, its basic uncertainty assignment will be:

$m_1(H) = 0$;
$m_1(\bar{H}) = \alpha$;
$m_1(U) = 1 - \alpha$.

Also, given the probability of the trustworthiness for driving distance evidence, we could construct its basic uncertainty assignment similarly. Note that we first decide driving fraud activities based on individual evidence. Also, instead of thresholding the coding cost (or anomaly degree), we simply decide top-10 activities with the top 10 highest coding cost (or anomaly degree) as frauds for each pair of source and end nodes, because the heterogeneous characteristics of all the trajectories makes it difficult to specify a common threshold.

The combined belief of travel route evidence and driving distance evidence in H is $bel(H) = m_1(H) \oplus m_2(H)$ following the Dempster's rule for the combination as shown in Equation 6:

$$\begin{aligned} bel(H) = m_1(H) \oplus m_2(H) = \frac{1}{C}\{m_1(H)m_2(H) \\ + m_1(H)m_2(U) + m_1(U)m_2(H)\} \end{aligned} \tag{7}$$

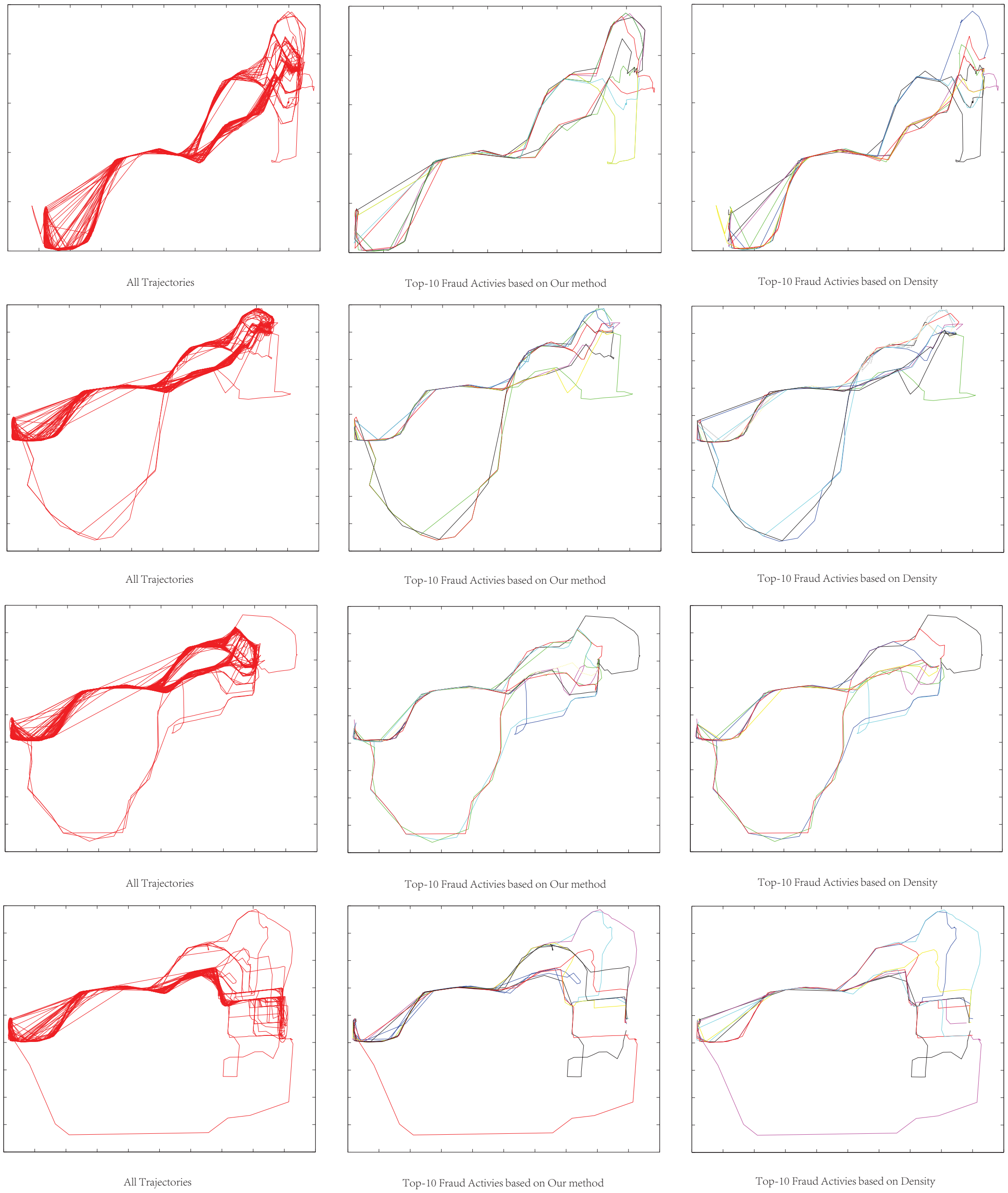
where

$$\begin{aligned} C = m_1(H)m_2(H) + m_1(H)m_2(U) + m_1(U)m_2(H) \\ + m_1(\bar{H})m_2(\bar{H}) + m_1(\bar{H})m_2(U) \\ + m_1(U)m_2(\bar{H}) + m_1(U)m_2(U) \end{aligned} \tag{8}$$

In addition, similar results can be obtained for $bel(\bar{H})$. In many applications, the basic probability of trustworthiness may not be available. One way to obtain this basic probability α is to look for cases in which it offers contradict evidence with the eventual judgment[14], [15]. Specifically, we estimate α by examining partial data. For example, if N trajectories are examined and M ones lead to contradict evidence to the final manual judgment, α is estimated as $(N - M)/N$.

VI. System Evaluation

In this section, we provide an empirical study of the taxi driving fraud detection system.



All Trajectories

Top-10 Fraud Activies based on Our method

Top-10 Fraud Activies based on Density

All Trajectories

Top-10 Fraud Activies based on Our method

Top-10 Fraud Activies based on Density

All Trajectories

Top-10 Fraud Activies based on Our method

Top-10 Fraud Activies based on Density

All Trajectories

Top-10 Fraud Activies based on Our method

Top-10 Fraud Activies based on Density

Figure 6. Fraud Driving Activities.

A. The Experimental Setup

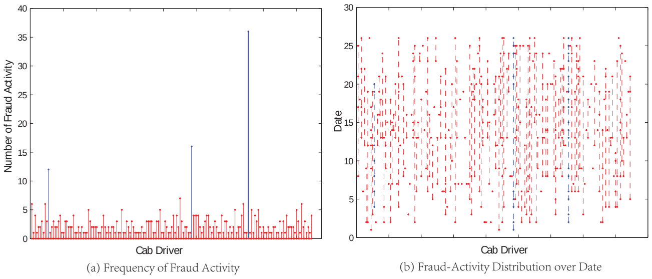
Experimental Data. We used real-world taxi GPS traces, which are provided by the cabspotting project [1]. This data set contains GPS traces of approximately 500 taxi cabs collected around 30 days in the San Francisco Bay Area. For each recorded point, there are four attributes: latitude, longitude, fare identifier and timestamp. Fare identifier is 1 if there are passengers within the cab or 0 otherwise. For the simplicity of computation and visualization, we first map latitude-longitude to 2-D flat coordinates.

Experimental tools and parameters. We use CLUTO [16] to perform clustering to identify 50 $interesting\ sites$. The size of grid is specified as 2. The threshold τ for $routemark$ construction is set as 0.85. Also, we maintain sequences of $routemark$, which represent more than 20% of trajectories from a source node to an end node. The common variance σ^2 of Gaussian for driving distance modeling is 0.1. The probability of trustworthiness for evidences m_1 and m_2 is estimated as $\alpha_1 = 0.85$ and $\alpha_2 = 0.9$ respectively.

B. Fraud Driving Activity

Here, we pick up 4 pairs of source and end nodes, and show the detected driving fraud activities by our Dempster-Shafer evidence combination in Figure 6 (Due to the limited space, more results are omitted). We return the top-10 driving fraud activities for each pair of source and end nodes. In the left column of Figure 6, we plot all trajectories from source node to end node. In the middle column, we show top-10 driving fraud activities based on our method. We used various colors to these 10 trajectories in order to make them differentiable. For comparison purposes, we also show top-10 driving fraud traces in the right column of Figure 6 by using density-based method. The basic idea of density-based method is to return the top-k trajectories which pass through grids with lowest density. Specifically, we calculate the density for each trajectory by averaging the density of all cells which are passed by this trajectory. By comparing the figures in the middle column to those in the right column of Figure 6, we can see our method based on DS evidence combination performs better than the density-based method.

After detecting the fraud activities for a pair of source and end nodes, we are able to count the number of driving fraud activities for each driver. In Figure 7 (a), we show the frequency of driving fraud activities for each driver within about 30 days. We highlight top-3 drivers, who commit the most frauds. To further observe if some drivers habitually commit driving fraud every day, we show the fraud activity day by day for each driver in Figure 7 (b), where each dot represents one fraud by a driver in one day. As can be seen, some drivers do have relatively habitual fraud behaviors, such as those top-3 drivers.



(a) Frequency of Fraud Activity

(b) Fraud-Activity Distribution over Date

Figure 7. Illustration for Fraud Activities.

To further understand the regularity of driving fraud activities, we explore the temporal distribution of all driving fraud activities in Figure 8, where the temporal unit is a hour. Each dot point represents a driving fraud activity with corresponding driver on x-axis and hour on y-axis. We highlight top-3 drivers with the most fraud activities as vertical blue lines and top-4 time zones (hours) containing most fraud activities as horizontal blue lines. Here, we can observe that more fraud activities are committed around 5PM, 7PM, 8PM and 10PM.

C. Mechanism for Possible Excuses

Here, we show additional functions of our system to avoid possible excuses by fraudulent drivers. As mentioned in section I, drivers may argue that they detour due to being unfamiliar with the roads or to avoid the traffic. To deal with these two possible excuses, we introduce one mechanism to our system. Specifically, after we detect a suspicious driving activity, we recall all previous driving traces of the corresponding driver to check if the driver is familiar with the roads. In parallel, to confirm the traffic-related excuse, we recall all driving traces around the same timestamp. For example, in Figure 9 (a), after detecting the driving fraud activity (red one), we recall and plot the driver's previous driving trajectories (green ones) and can clearly find this driver often operates in this area. Also, in Figure 9 (b), we show all trajectories (green ones) happening within the same minute as the driving fraud activity (red one) and we can see many other drivers did not detour. By this mechanism, we are able to obtain more real-time evidence to deny possible excuses and confirm the fraud behaviors.

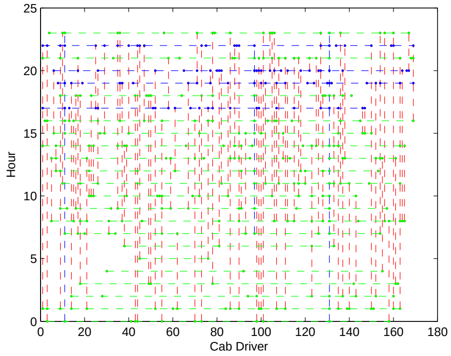


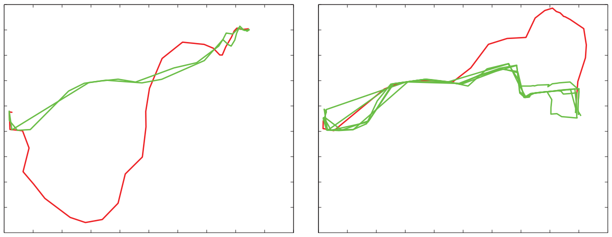
Figure 8. Temporal Distribution of Fraud Activity.

D. The Motivation Analysis

Here, we analyze the driving fraud activity data and investigate the motivation behind the fraud. Specifically, we investigate the day income of taxi drivers.

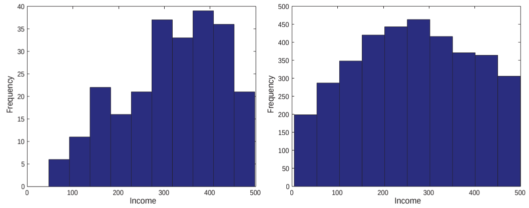
First, let us introduce the data for this study. In total, we obtain 436 driving fraud activities. For each fraud activity, we record three attributes: $DriverID$, $Date$, and $DayIncome$. These records are denoted as F. Note that the same driver may commit more than one fraud one day. Then, we further get another set of records (denoted as A) from the meta data. Each record contains attributes: $DriverID$, $Date$, $DayIncome$ and $Indicator$. $Indicator$ is 1 if this driver commits a driving fraud on the corresponding date, and 0 otherwise. In total, we have 4943 records in A. From A, we select partial records with $DayIncome$ equal to or less than the maximum $DayIncome$ in F. These partial records are denoted as P. Then, we separate P into 2 groups: FP and NP. FP is the group of 242 records with fraud and NP is the group of 3617 records without fraud.

We show the histogram of day income of FP and NP in Figure 10. We assume that the income of NP follows normal distribution and we estimate the mean $\mu_n = 263.6$ and standard deviation $s_n = 129.0$ from the records of NP. Then, we test whether the average income $\mu_f = 316.5$ of FP is different from μ_n. With standard two tailed



(a) (b)

Figure 9. The Mechanism for handling Excuses.



(a) Income Hist of Drivers with Fraud (b) Income Hist of Drivers without Fraud

Figure 10. Income Histogram.

hypothesis test, we find μ_f is significantly different from μ_n with a significant level as 0.01. From this statistic test, we can see that the fraudulent drivers averagely earn more income than drivers without fraud. Thus, by committing frauds, fraudulent drivers do earn more income than other drivers with statistical significance.

VII. Related Work

In general, related work can be grouped into three categories. The first category includes the work on trajectory outlier detection, which is highly related to this work. For instance, Lee et al. [7] proposed a two-phase trajectory partition strategy for detecting trajectory outliers. This work has exploited both distance and density information for outlier detection. Also, in [17], an outlier detection framework was proposed for monitoring anomalies over continuous trajectory streams. The key idea is to built local clusters upon trajectory streams and detected anomalies by a cluster join mechanism. Finally, an incremental semi-supervised learning method was developed in [18] for trajectory outlier detection. This work is along the line of a learning approach and requires the training data. In a nutshell, while various aspects of abnormality of trajectory data have been exploited, most methods can not direclty be used to the specific taxi driving fraud detection due to serveral unique characteristics of this problem, such as multiple evidences, possible excuses. Furthermore, we proposed a DS theory-based method to combine the multiple evidences, which has not be studied in most previous works about trajectory outlier detection.

The second category includes the work on more general analysis of trajectory data, such as trajectory clustering and trajectory pattern mining. For instance, Giannotti et al.[19] introduced trajectory patterns as concise descriptions of frequent behaviors in terms of both time and space. Also, a trajectory clustering algorithm was proposed in [20]. This clustering algorithm first partitions the trajectories according to the Minimum Description Length principle and then clustered the trajectory segments using a line-segment clustering algorithm. In [21], a filter-refinement approach was developed for discovering convoys in trajectory databases. Moreover, people have various interests in developing similarity and distance measures for trajectories [22], [23]. Finally, in [24], Wang et al. proposed to co-cluster trajectories and semantic regions with a Bayesian model called Dual Hierarchical Dirichlet Process by treating trajectories as documents and positions as words. All the above mentioned works on location traces have shed the light on the taxi driving fraud detection problem studied in this paper, while these works target completely different problems.

The third category includes outlier detection methods that are not designed for trajectory data. For example, Wu et al. proposed an algorithm to find the rectangular regions where data exhibit anomalous behaviors [25]. Also, an angle-based outlier detection algorithm was proposed for high-dimensional data [26] and a parameter-free outlier detection method [27] was developed by exploiting the concept of coding cost in information theory. Finally, people are also interested in developing temporal outlier detection methods for data stream. As an example, Pokrajac et al. introduced an incremental local outlier detection method for data streams [28]. They adapted Local Outlier Factor (LOF) [29] for incrementally detecting outliers in data stream.

VIII. Conclusion

In this paper, we developed a taxi driving fraud detection system. To implement this system, we exploited a large amount of GPS traces collected from about 500 taxi drivers and systematically examined the taxi driving fraud activities. Specifically, we mainly considered two aspects of evidences: travel route evidence and driving distance evidence. To discover the travel route evidence, we encoded the trajectories via $symbol$ and oversampling. Also, considering the natural correlations among generated $symbols$, we applied ICA for extracting independent components before encoding the trajectories. In addition, we introduced the notion of $routemark$ to detect a typical driving path from one interesting site to another one, and then statistically modeled the driving distance by using the identified routemarks. The driving distance evidence is derived by the statistical model. Finally, we effectively combined these two aspects of evidences based on the Dempster-Shafer theory and obtained more robust evidences for detecting driving fraud activities.

Finally, we conducted extensive experiments with real-world taxi GPS logs to show the effectiveness of the taxi driving fraud detection system. Along this line, we discovered some regularities of taxi driving fraud activities, such as the temporal distribution of driving frauds. Also, we investigated an interesting motivation of taxi drivers to commit a driving fraud. Indeed, the development of this taxi driving fraud detection system provides a new paradigm for understanding the taxi driving fraud activities and obtaining more guidance to deal with the taxi fraud.

IX. Acknowledgements

This research was supported in part by National Science Foundation (NSF) via grant numbers CCF-1018151 and IIP-1069258, National Natural Science Foundation of China (NSFC) via project numbers 70890082, 71028002, 60975043, and 61073097, 973 Program (2010CB327903) of China, and the Fund of Ministry of Education of China (10YJC630065).

References

[1] “http://www.consumertraveler.com/today/nyc-taxi-drivers-overcharge-passengers-8-3-millioin/.”

[2] “http://www.tour-beijing.com/taxi/.”

[3] “http://www.bustathief.com/taxi-fraud-taxi-scam/.”

[4] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, “An energy-efficient mobile recommender system,” in *ACM SIGKDD*, Washington D.C., 2010.

[5] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, and Y. Huang, “T-drive: Driving directions based on taxi trajectories,” in *ACM SIGSPATIAL GIS 2010*, San Jose, California, 2010.

[6] Y. Ge, C. Liu, and H. Xiong, “A taxi business intelligence system,” in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, US, 2011.

[7] J.-G. Lee, J. Han, and X. Li, “Trajectory outlier detection: A partition-and-detect framework,” in *IEEE ICDE*, Cancun, Mexico, 2008, pp. 140–149.

[8] Y. Ge, H. Xiong, Z.-H. Zhou, H. Ozdemir, J. Yu, and K. Lee, “Top-eye: Top-k evolving trajectory outlier detection,” in *ACM CIKM*, Toronto, CA, 2010.

[9] V. Pless, *Introduction to the Theory of Error-Correcting Codes*. John Wiley Sons, ISBN: 0-471-08684-3.

[10] D. E. Knuth, “Dynamic huffman coding,” *Journal of Algorithms*, vol. 6(2), pp. 163–180, 1985.

[11] J. Stone, “A brief introduction to independent component analysis,” *Encyclopedia of Statistics in Behavioral Science*, vol. 2, pp. 907–912, 2005.

[12] H. A, “Fast and robust fixed-point algorithms for independent component analysis,” *IEEE Transactions on Neural Networks*, vol. 10(3), pp. 626–634, 1999.

[13] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[14] Y. Jin, L. Khan, L. Wang, and M. Awad, “Image annotations by combining multiple evidence & wordnet,” in *Proceedings of the 13th annual ACM international conference on Multimedia*, Singapore, 2005.

[15] T. M. Chen and V. Venkataramanan, “Dempster-shafer theory for intrusion detection in ad hoc networks,” *Journal of IEEE Internet Computing*, vol. 9(6), pp. 35–41, 2005.

[16] G. Karypis, “Cluto: http://glaros.dtc.umn.edu/gkhome/views/cluto.”

[17] Y. Bu, L. Chen, A. W.-C. Fu, and D. Liu, “Efficient anomaly monitoring over moving object trajectory streams,” in *ACM SIGKDD*, Paris,France, 2009, pp. 159–168.

[18] R.R.Sillito and R.B.Fisher, “Semi-supervised learning for anomalous trajectory detection,” in *Proceedings of the 19th British Machine Vision Conference*, Leeds, UK, 2008, pp. 1035–1044.

[19] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, “Trajectory pattern mining,” in *ACM SIGKDD*, California, USA, 2007, pp. 330 – 339.

[20] J.-G. Lee, J. Han, and W. Kyu-Young, “Trajectory clustering: a partition-and-group framework,” in *ACM SIGMOD*, Beijing, China, 2007, pp. 593–604.

[21] H. Jeung, M. L. Liu, X. Zhou, C. S. Jensen, and H. T. Shen, “Discovery of convoys in trajectory databases,” in *Proceedings of the VLDB Endowment*, 2008, pp. 1068–1080.

[22] L. Chen, M. T. Ozsu, and V. Oria, “Robust and fast similarity search for moving object trajectories,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, IL, USA, 2005, pp. 491 – 502.

[23] M. Vlachos, D. Gunopoulos, and G. Kollios, “Discovering similar multidimensional trajectories,” in *Proceedings of the 18th International Conference on Data Engineering*, CA, US, 2002, pp. 673–684.

[24] X. Wang, K. Ma, G.-W. Ng, and W. E. L. Grimson, “Trajectory analysis and semantic region modeling using nonparametric bayesian model,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Patter Recognition*, AK, USA, 2008, pp. 1–8.

[25] M. Wu, X. Song, S. R. Chris Jermaine, and J. Gums, “A lrt framework for fast spatial anomaly detection,” in *ACM SIGKDD*, Paris,France, 2009, pp. 887–896.

[26] H.-P. Kriegel, M. Schubert, and A. Zimek, “Angle-based outlier detection in high-dimensional data,” in *ACM SIGKDD*, Nevada, USA, 2008, pp. 444–452.

[27] C. Bohm, K. Haegler, N. S. Miller, and C. Plant, “Coco: Coding cost for parameter-free outlier detection,” in *ACM SIGKDD*, Paris, France, 2009, pp. 149–158.

[28] D. Pokrajac, A. Lazarevic, and L. J. Latecki, “Incremental local outlier detection for data streams,” in *IEEE Symposium on Computational Intelligence and Data Mining*, Hawaii, USA, 2007, pp. 504–515.

[29] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: Identifying density-based local outliers,” in *ACM SIGMOD*, Texas,US, 2000, pp. 93 – 104.