Detecting Pickpocket Suspects from Large-Scale Public Transit Records

***Kaviraj**, Student Dept. of ISE, The National Institute of Engineering, Mysuru, Karnataka, India*

Abstract -*Massive data collected by automated fare collection (AFC) systems provide opportunities for studying both personal traveling behaviors and collective mobility patterns in urban areas. Existing studies on AFC data have primarily focused on identifying passengers' movement patterns. However, we creatively leveraged such data for identifying pickpocket suspects. Stopping pickpockets in the public transit system has been crucial for improving passenger satisfaction and public safety. Nonetheless, in practice, it is challenging to discern thieves from regular passengers. In this paper, we developed a suspect detection and surveillance system, which can identify pickpocket suspects based on their daily transit records. Specifically, we first extracted a number of useful features from each passenger's daily activities in the transit system. Then, we took a two-step approach that exploits the strengths of unsupervised outlier detection and supervised classification models to identify thieves, who typically exhibit abnormal traveling behaviors. Experimental results demonstrated the effectiveness of our method. We also developed a prototype system for potential uses by security personnel.*

Key Words: —Automated Fare Collection, Travel Behaviors, Mobility Patterns, Public Safety, Anomaly Detection

1. INTRODUCTION

Public transit passengers can easily become distracted in crowded environments, where they are often rushing from one location to another. Having their focus drift from their belongings, they often become common targets of pickpockets [1, 2]. During the first 9 months of 2014, it was reported that 350 pickpockets were apprehended in the subway system and 490 on buses in Beijing.1 Many other big cities around the world, such as Barcelona, Rome, and Paris, also suffer from pickpocket problems.2 Indeed, it is challenging to detect theft activities committed by cunning thieves who know how to escape without being discovered. It is critical to provide a smart surveillance and tracking tool for transit system security personnel. With rapid advances in information technology and infrastructure, transactional records collected by automated fare collection (AFC) systems are now available for understanding passengers' mobility patterns and urban dynamics [3, 4, 5, 6, 7]. Most existing studies focus on identifying regular, collective mobility patterns, such as commute flows and transit networks. Our study is the first to focus on identifying thieves based on AFC data. It is possible to detect thieves using AFC records because behavioral differences logged in the mobility footprints may be used to separate suspects from regular passengers. Examples of such behaviors include traveling for an extended length of time, making unnecessary transfers, and taking regular routes with random stops. Designing an intelligent system that automatically extracts specific, identified behavioral features, and dynamically detects and tracks pickpocket suspects has become a possibility. Detecting thieves based on AFC records is not a simple outlier detection problem. Fig. 1 shows the difference between a known thief and an outlier. We can see a number of trajectories between hot regions A and B. By careful examination, we see that most passengers move from one region to another using a near-optimal configuration (e.g., shortest time/distance, or a minimal number of transfers). However, a passenger (a known suspect) who took the path A → C → D → B looks suspicious because there is no need to make transfers at C and D in order to reach B. Based on the above observation, passengers who exhibit such abnormal behaviors will be selected for further examination. In contrast, another passenger who travels from E to B is an outlier, since few passengers take the same path. However, this passenger is likely just a regular passenger who originates from a less crowded area. Detecting thieves is challenging also because not every trip made by a regular passenger looks normal. Regular commuters may occasionally make trips to visit friends or places of interest, and such trips may look suspicious by how much they deviate from regular passenger behaviors. Adding to this complex landscape, a large number of AFC records are being collected from millions of passengers, when only a tiny fraction of passengers are actual pickpockets. Pinpointing such a small group of people within such a large-scale dataset is analogous to searching for a needle in the haystack. Meanwhile, we need to effectively transform our knowledge based on model development into a decision support system. Such a system needs to provide real-time decision recommendations to guide security personnel to perform their work more efficiently. In this paper, we adopted a comprehensive approach to the pickpocket detection problem. The overall framework of our solution is illustrated in Fig. 2. We first partitioned

the city area into regions with functional categories. Then, the mobility characteristics of passengers were extracted from transit records dynamically over time. A core component of the system was a two-step passenger classification process, the first step being regular passenger filtering, and the second step being suspect detection. Finally, system user feedback information, such as newly confirmed thieves, was entered as ground truth for future model training. The two-step pickpocket detection framework was proposed to combat the problem of extreme imbalance between positive and negative samples. In our preliminary work [8], we assessed the feasibility of this two-step framework with promising results. A major difference in this current study from our preliminary work is that instead of using offline data, we designed a real-time system with a dynamic update mechanism. Specifically, an ensemble of many base models was employed in the regular passenger filtering step, which helped with balancing training samples for an improved degree of generality and better suspect detection performances. The base models in this ensemble system were managed with a dynamic updating mechanism, which was provided based on a utility function that strikes a tradeoff between effectiveness (i.e., performance) and relevance (i.e., recency). A more detailed description of this system may be found in Section 5.2. The contribution of our study can be summarized as follows. Firstly, we identified a number of features that may be extracted from AFC records and are potentially useful for distinguishing thieves from regular passengers. Secondly, a two-step approach was proposed to make the suspect detection problem practical in a large-scale data environment where the positive and negative samples are extremely imbalanced. Thirdly, our dynamic filtering enhancement significantly reduced the everyday computation costs and maintained superior accuracy. Most importantly, a real system for the end user was designed and tested using real-world, large-scale data. As an applied data science study, our solution is the first to address an important social issue—identifying pickpockets—by using big data. The significance of this work has been recognized by a featured article in The Economist. 3 The remainder of this paper is organized as follows. First, we will summarize related work in Section 2. Then, we provide an overview of the real-world datasets in Section 3. A detailed description of features that we extracted to characterize mobility profiles of passengers is presented in Section 4. A two-step framework of the suspect identification system is proposed in Section 5. Details of the real-time system design are presented in Section 6. , we also provide an overview of the deployed system in Section 7. After presenting and discussing experimental results in Section 8, Finally, we draw conclusions in Section 9.

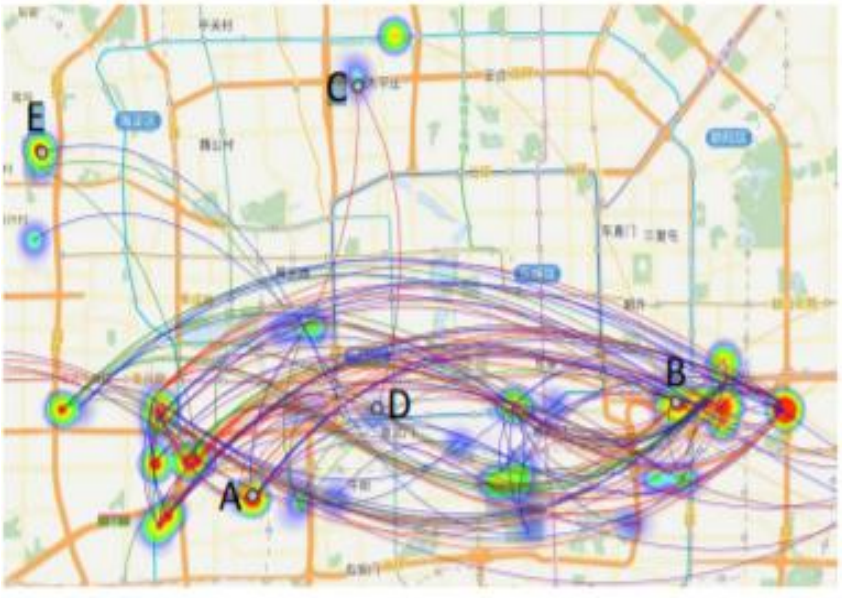


Fig. 1: Example trajectories of passengers.

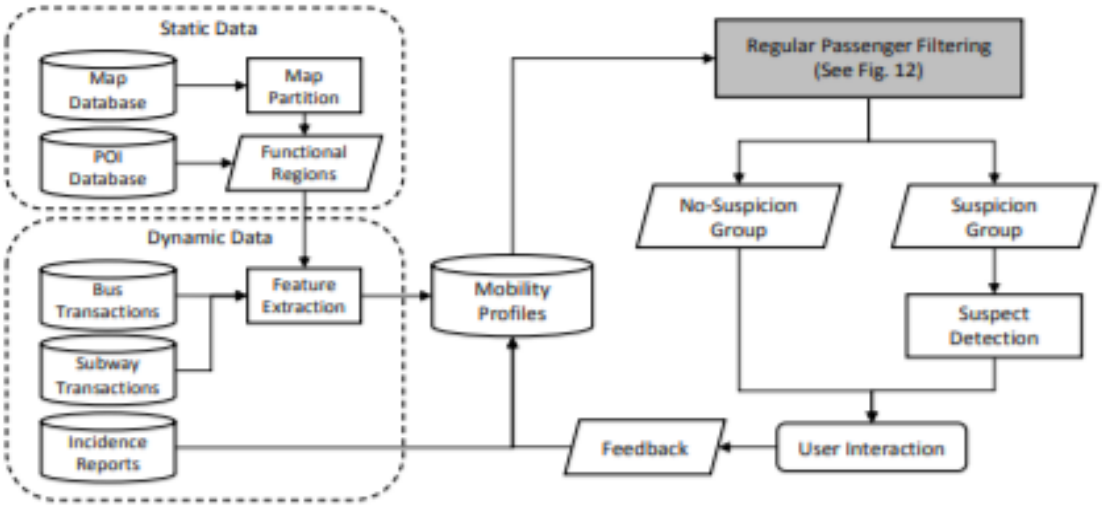


Fig. 2: The overall framework.

2. RELATED WORK

As urban sensing data, such as GPS traces, call detail records, and smart card logs, grow ubiquitous, research efforts devoted to analyzing such data have resulted in a number of works in recent years. In the context of mining public transportation data, in this section, we provide a brief review of the related work.

2.1 Passengers Activity

Patterns the first group of existing literature focuses on finding patterns in passenger activity records. Such knowledge can be useful in a variety of applications, and plays a vital role in effectively finding and satisfying passenger needs. Examples include assessing the performance of the transit network, identifying and optimizing problematic or flawed bus routes, improving the accuracy of passenger flow forecasted between two regions, and making service adjustments that accommodate variations in ridership on different days. In particular, [4] estimated the crowdedness of various stations in the transportation network using AFC data. [9] measured the variability of transit behaviors on different days of the week. In addition, different studies have investigated unique characteristics of traveling patterns of the elderly [10], students, and adults [9], which provided interesting insights for understanding behavioral differences of sub-populations. It has been suggested that human mobility patterns follow a high degree of spatial and temporal regularity, and are thus highly predictable [11, 12]. By identifying trip patterns, these studies typically aimed to discovering movement patterns by finding frequently visited places of regular passengers, who traveled the same sequence of places at a similar time of day. For example, [13] identified spatiotemporal patterns from GPS traces of taxis for night bus route planning. [14] tried to predict the most common routing preference of past passengers by identifying the most frequented travel paths during a certain time period. [3] analyzed sets of moving objects, like traffic patterns, bird migration, and infectious disease transmission, to discover and explain movement patterns.

2.2 Abnormal Traveling Behavior Detection

Existing studies that detect anomalies in urban sensing data can be divided into two categories: those based on locations, and those on trajectories. Along the line of location-based anomaly detection, [15] presented a framework that learned the context of different functional regions in a city, which provided the basis of our feature extraction approach (see Section 3.2). In addition, [16] attempted to discover casual relationships among spatiotemporal outliers. [17] mined representative terms from social media posts when location-relevant events happened in the city, such as accidents or protests. [18] discovered black-hole or volcano patterns in human mobility data in a city, which could quickly identify gathering events, such as football matches or concerts. Detection of such anomalies can help send alerts, and provide input for intelligent decision support, such as smoothing the

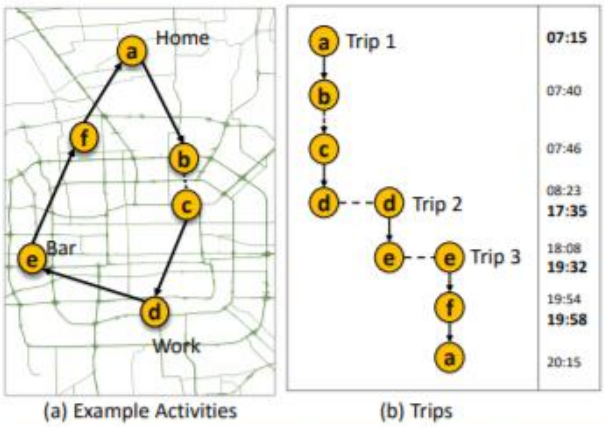
traffic flow [18]. The main goal of trajectory based anomaly detection is to discover a small percentage of individuals, whose movement traces are uniquely different from the general population. One example is to identify fraudulent taxi driving behaviors. A large number of studies have investigated trajectory based anomaly detection using data mining techniques, such as graph based [3], clustering based [4, 5, 19], local/context-aware based [20, 21], dimension reduction based [22], and evidence based (e.g., using Dempster-Shafer theory [23]). While the trajectory of pickpockets containing features that are implicit, previously unknown, and potentially useful from large datasets, pickpocket suspect detection based on AFC records is a novel problem that has not been considered in the literature, and proves to be a challenging research endeavor.

3. DATA DESCRIPTION

The data for our study were collected from multiple sources. These include transit records, geographical information, and theft incident reports. In this section, we provide an overview of the data.

3.1 Transit Records

Our study is based on a large-scale transit records dataset collected from a public transit system that includes buses and subways. Passengers utilizing the transit service are charged by the distance they travel. A rechargeable smart card is issued to each passenger, who swipes the card when they boarding or exiting a vehicle. The AFC system then calculates the fare according to the stations of boarding and exiting. As a result, each raw AFC record consists of the smart card ID, the route number, the event (i.e., boarding or exiting), the station, and the time stamp. We transformed the data so that each transit record consists of one boarding and one exiting event of the same ID. In order to describe the data and subsequent feature extraction process clearly, we first clarify two concepts, transit records and trips, by providing a concrete example. Fig. 3 illustrates an example passenger's activities in a typical day. Part (a) is the actual trajectory on the city's map; Part (b) splits the trajectory into three separate trips; and Part (c) demonstrates the corresponding transit records in our data. Specifically, Passenger 4322 left home via Route 52 at Station a at 7:15 a.m. At 7:40 a.m., he transferred by exiting Route 52 at Station b and walking across the street to take Route 26 at Station c (7:46 a.m.). Then he exited at Station d, next to his workplace, at 8:23 a.m. We determined that he completed Trip 1 because the next time he entered the transit system was more than 30 minutes later (i.e., our empirical cutoff). Therefore, the afternoon transit from d to e was considered Trip 2; and the transit from e back home to a, with a transfer at f, was considered Trip 3. As a result, we collected five transit records for Passenger 4322 that describe three trips.



(a) Example Activities (b) Trips

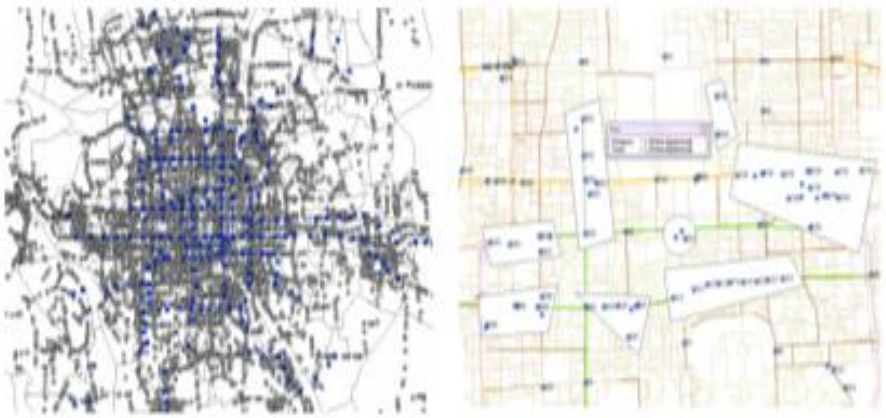
Smart Card ID	Route Number	Boarding Station	Boarding Time	Exiting Station	Exiting Time
4322	Route 52	a	07:15	b	07:40
4322	Route 26	c	07:46	d	08:23
4322	Route 11	d	17:35	e	18:08
4322	Route 11	e	19:32	f	19:54
4322	Route 16	f	19:58	a	20:15

(c) Transit Records

Fig. 3: An example of trips and transit records.

Intuitively, a transit record corresponds to one segment of a passenger's transit between a pair of consecutive boarding and exiting events. Even though this segment of transit may pass a number of stations, the passenger does not exit the vehicle during this time. In contrast, a trip consists of one or more such segments, which connect places of interest (i.e., where the passenger stays for extended periods of time) on the two ends. A trip may include connections or transfers, as long as those breaks are relatively short (e.g., 30 minutes or less) in time. Formally, we provide the following definitions for concepts that are important for this study. Definition 1 (Transit record). A transit record tr contains the following information: • trroute: the bus/subway route number; • trsboard, trtboard: the boarding station and time; and • trsexit, trtexit: the exiting station and time. As a result, for each transit record, we were able to compute the travel distance trdist, travel time trtime, and number of stops trstops during the transit. Definition 2 (Trip). A trip T r is a sequence of transit records T r = (tr1 , tr2 , . . . , trn), where the passenger's origin location is T rorigin = tr1 vboard and the destination is T rdest = trn sexit. In practice, we construct one trip record if and only if the time gap between two consecutive transit records is 30 minutes or less. The trip's time duration is calculated as "T rtime = trn texit − tr1 tboard".

The public transit network dataset provides the geocoordinates of the various bus and subway stations on the road networks. As shown in Fig. 5(a), in total, we have 44,524 bus stations (points in gray) covered by 896 bus routes, and 320 subway stations (points in blue) covered by 18 subway routes. To remove the redundancy and better model the mobility patterns, we merged stations located at the same road intersection, as demonstrated in Fig. 5(b).



(a) Stations (b) Clustered Locations

Fig. 5: Stations information.

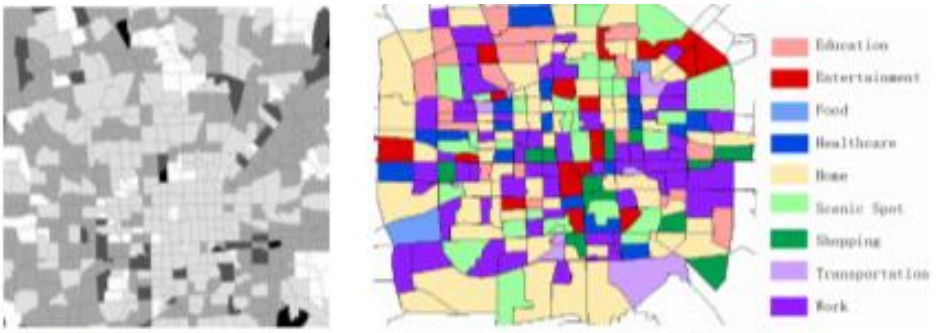
3.2 Geographical Information

To project the transit records to the city map, we made use of external datasets of important geographical information, namely, the road network data, the points of interests (POI) data, and the public transit stations. The POI data consist of the geo-coordinates of businesses and landmarks with their categories. We consider ten broad categories of POI, as summarized in Table 1.

TABLE 1: Categories of POI and frequencies.

Category	Examples	Frequency
Home	Apartment buildings	28,731
Work	Office buildings	71,364
Education	Schools, training centers	3,527
Food	Restaurants and dining	56,906
Shopping	Shopping malls and outlets	24,310
Entertainment	Museums, theaters, clubs	18,223
Scenic Spot	Parks, sports fields	2,362
Transportation	Airports, transit centers	15,287
Healthcare	Hospitals, pharmacy	8,685
Car services	Car sales, repairs	1,781

As a preprocessing step, we first followed Yuan's work [15] to segment the urban area into small regions by major road networks, as shown in Fig. 4(a). Then, using the POI data, we categorized each region into one of the ten functional zones we identified in Table 1. These regions are then color coded visualized in Fig. 4(b)



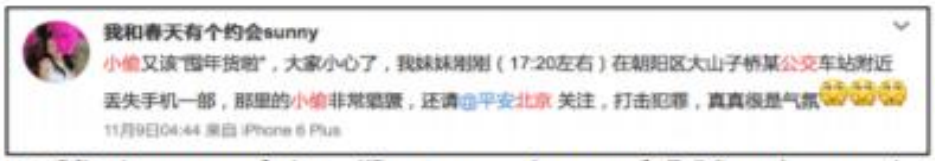
(a) Map Segmentation (b) Functional Regions

Fig. 4: Geographical information.

3.3 Incident Reports

Confirmed pickpocket incidents are publicly announced via Sina Weibo, the primary social network service in China. It is considered public data since all posts are visible to everyone, just like Twitter in the United States. By searching a number of keywords (e.g., Beijing, public transportation, and thief), we initially identified 10,529 records of Weibo posts during our study period. We included two types of pickpocket reports: official announcements posted by the police4 , and personal complaints posted by the victims. Fig. 6 provides an example of each type of report. The date, time, and location of the theft events are normally identified in the posts, which enabled our ability to link such events to other sources of data.

(a) Police report: "At 7:40 a.m. on July 10th, a thief was caught at Route 349 East Chengzhuanglukou Station."



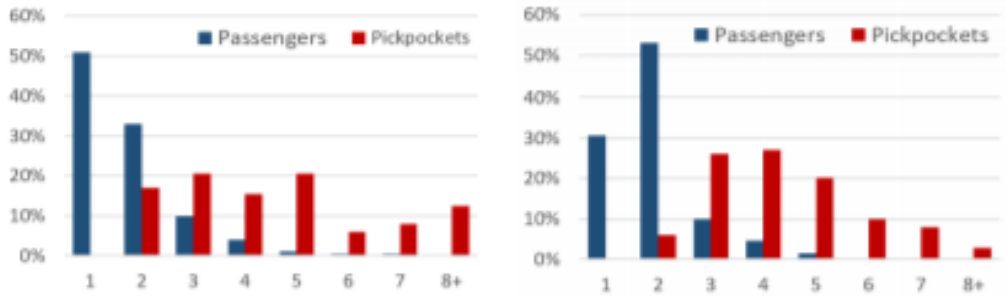
(b) Victim complaint: "Just now (around 5:20pm), my sister's phone was stolen at the Dashanzi Bridge Bus Station."

Fig. 6: Example incident reports on Weibo.

Among the Weibo posts found, we were able to identify 873 theft events according to route or station information with matching date and time stamp. For police reported events, thieves were caught and remove from the vehicle. We were able to uniquely determine their card ID since their entry to the transit system was documented but their exit was not. Oftentimes these same smart cards were no longer active for the next 48 hours. We flagged them as actual thieves. Since thieves involved in victim reported events were not caught, we could not identify them according to the no-checkout rule. Instead, we manually labelled thieves according to their travel behaviors. Specifically, we first identified all passengers on the vehicle during the same period of time, and then visualized their trajectories to ascertain whether their travel patterns were typical. This was a manual process performed by a public transportation expert who was familiar with the city. Some events might be associated with more than one thief. For example, the police may report catching several thieves who commit crimes as a group. Therefore, from these 873 police and victim reported events, we distilled 936 card IDs and flagged them as as true suspects.

4. MOBILITY CHARACTERISTICS

To distinguish pickpocket suspects from regular passengers, we extracted a number of features from passengers' AFC records. In this section, we describe these features and discuss their potential use for characterizing travel patterns in the public transit system. 4.1 Travel Time and Frequency The daily travel time is defined as the total duration spent by each passenger in the public transit system. The daily riding frequency is defined as the number of transit records traveled by each passenger per day. Indeed, pickpocketing is hard work: a thief has to spend a long time in crowded buses, subways, or stations to identify easy targets. To find more theft opportunities, a pickpocket tends to stay in the public transit system for a long time and makes random, frequent transfers.



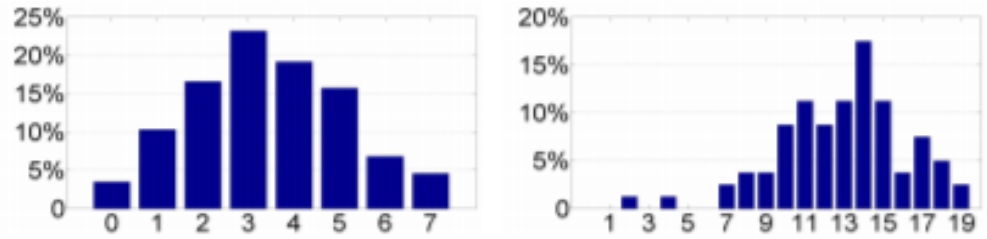
(a) Travel time (hours). (b) Riding frequency.

Fig. 7: Distributions of travel time and frequency.

Fig. 7 plots the distribution of daily travel time and riding frequency, respectively. We can see that more than 80% of passengers finish their travels within 2 hours and within 2 transit records per day. In comparison, the identified thieves often spend more than 3 hours traveling daily, with a higher daily riding frequency.

4.2 Short-Distance Rides

A short-distance ride is considered a transit segment that passes less than three station stops. Due to the density of stations in Beijing, normal passengers usually take rides that pass a larger number of stations. In contrast, pickpockets often switch routes within a few station stops to avoid attracting fellow passengers' attention and being recognized.



(a) With 7 Transit Records (b) With 19 Transit Records

Fig. 8: Distributions of passengers with short-distance trips.

Fig. 8 shows the distribution of passengers taking various numbers of short-distance rides, for those with 7 and 19 transit segments, respectively. In both plots, the x-axis is the number of short-distance rides and the y-axis is the percentage of passengers. For passengers with 7 transit segments, as shown in Fig. 8(a), the distribution is approximately Gaussian with mean 3. For those with 19 transit segments, as shown in Fig. 8(b), the peak of the distribution is shifted to the right, showing that the relative frequency of short distance rides increases.

4.3 Transitions among Functional Zones

A high-level view of human mobility patterns can be summarized as transitions among regions, where each region may cover multiple stations. For example, the morning commuting trips can be abstracted as "departing from residence region, then transferring at a transit hub, and finally arriving at a business region." Such sequential information proves useful for understanding mobility patterns and predicting traffic flows [24, 25, 26]. Comparing regular passengers and pickpockets, we observed that regular passengers often presented typical sequential

patterns, whereas pickpockets were more likely to wander randomly among the city's functional regions. Features to represent transitions among functional zones were extracted as follows. First, we preprocessed the geographical information, as described in Section 3.2. Then, each station s ∈ S were labeled by the region r containing the station, denoted by s ∈ r. For each transit record tr, we also denoted tr ∩ r 6= ∅ if trsboard ∈ r or trsexit ∈ r. As a result, we were able to define features such as number of boarding stations and number of boarding regions. Next, we counted the transition frequency between any pair of function categories for each passenger's daily trips. For M functional categories, the functional transition features would be represented as a matrix F T ∈ RM×M. Fig. 9 shows an example of a passenger trip, the corresponding transition matrix, and the vector of transition features, where R, T, SC, and S represent residence, transportation, scenic spot, and shopping, respectively. In this example, only these four functional categories are considered. Specifically, the passenger in Fig. 9(a) left one residence location, and after lingering around three other locations with the same function, took a round trip passing one transportation location, one scenic spot location, and one shopping location before returning to the initial residence location. The corresponding transition matrix of the trip is shown in Fig. 9(b). After vectoring this frequency matrix, we can represent this trip's functional transition features as shown in Fig. 9(c).

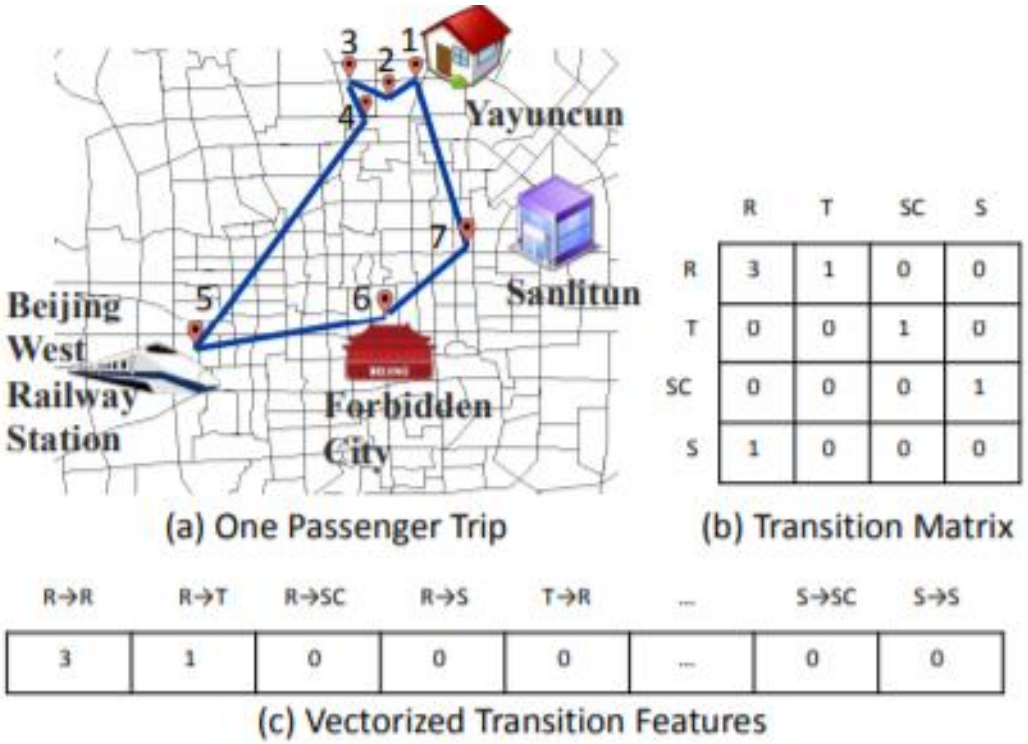


Fig. 9: Transitions among functional regions.

4.4 Frequently Visited Regions

Most people visit a small number of familiar locations. Likewise, pickpockets often spend a large portion of time within a few routes or regions. For example, some thieves prefer to scout targets in busy transit hub stations, and follow them onto buses or trains. Once the crime has been committed or the target lost, the thief would likely continue to look for the next victim. Thieves are intimately familiar with these areas and wander around them. We can measure their wandering behaviors to identify suspects with simple variables like the maximum number of times a route was taken or the maximum number of visits made to a region. These wandering behaviors cluster around passengers' boarding or exiting locations.

Fig. 10: Spatial clustering characteristics.

A density-based algorithm like DBSCAN [27] could be executed to find these location clusters. The DBSCAN results for the example passenger (an actual pickpocket) shown in Fig. 10, where each square represents a boarding or exiting event, the thief visited 38 boarding and exiting locations, and 11 clusters were formed. We could then use the number of clusters located as a feature to measure the extent of wandering concentration.

4.5 Deviation from the Social Norm

A group of features were considered to represent deviations from normal behaviors of most passengers. These are labelled as social features and help determine deviations from the social norm. Two very informative social features are the time gap of trips and the time gap of region transitions. Given the same origin and destination, the trip variation of the majority of the population (i.e., regular passengers) is low. For example, most of the trips will be finished within a specific amount of time given the trip origin and destination, while pickpocket suspects may spend more time in the transit system during the trip. Thus, for each origin station and destination station pair, we find the distribution of travel time by all trips passing this pair. We then convert each passenger's travel time into a significance score by the quantile in the distribution. Similarly, we also define the time gap significance of region transitions.

4.6 Historical Behaviors

We compute the statistics (e.g., median and standard deviation) of the daily features observed over the last 30 days for each passenger to quantify their historical behaviors. We use the median instead of the mean because the median is more robust in the presence of outliers, such as non-routine passenger behaviors. The standard deviation indicates the degree of variation of the daily behaviors of individual passengers. Regular passengers following routine trajectories normally generate statistics with less variation. Moreover, we also record the number of days (out of the recent seven days) when a passenger was detected as a potential pickpocket suspect. All features are summarized in Table 2. The statistics are calculated based on one-month's data. We first found the values of all features of each day, and then summarized the mean and median by weekdays and weekends.

5. SUSPECT IDENTIFICATION

This section presents the key component of our thief footmark detection system. Specifically, to distinguish pickpockets from the regular passengers with high accuracy and low false-positives, we developed a two-step framework. 5.1 A Two-Step Framework The rationale for using a two-step framework for identifying pickpocket

suspects is as follows. On the one hand, since the majority of the passengers are not thieves, it is impractical to use a classification algorithm. Specifically, the percentage of confirmed pickpockets is extremely low in the passenger population. Simple heuristics like oversampling and under-sampling would only be helpful for handling moderate class imbalance, but not an imbalance as extreme as ours. Building robust machine learning models for such unbalanced data is still an active research area in the literature [28, 29, 30, 31]. Alternatively, utilizing anomaly detection algorithms, which are typically unsupervised, cannot scale well and may also lead to significant false positives, since many regular passengers who occasionally perform irregular activities may be misclassified as suspects. To address these challenges, we develop a two-step framework by unifying the unsupervised anomaly detection and supervised classification in a novel way. We show that the two steps can effectively utilize the supervised information, overcome the issue of an unbalanced class distribution, and reinforce the learning performances. The overall framework can predict the pickpockets with very low false-positives. In the two-step framework, we approximate the predictive function f(·) as f(x) = g(x)h(x), and

equivalently:

$$f(x) = \begin{cases} 0, & \text{if } g(x) = 0; \\ h(x), & \text{if } g(x) = 1. \end{cases} \quad (1)$$

This relationship is also demonstrated in Fig. 11.

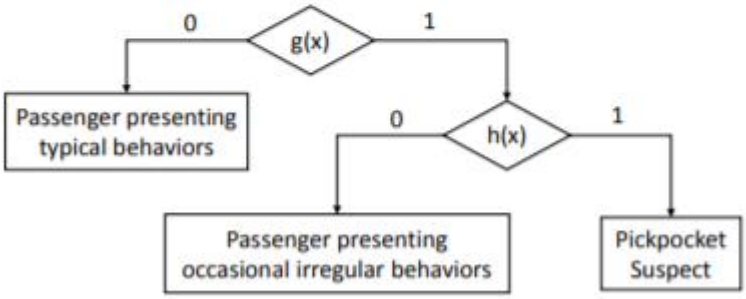


Fig. 11: A two-step approach for suspect detection.

In other words, we first use function g(·) (the first step) to filter out regular passengers whose mobility patterns are typical of the majority of the passenger population. If the passenger associated with feature vector x is not filtered out (i.e., g(x) = 1), we then use function h(·) (the second step) to detect whether the passenger is a pickpocket suspect. In the following section, we develop the two steps (function g(·) and h(·)) in our framework.

TABLE 2: List of extracted features.

Description	Weekends Mean	Weekends Med.	Weekdays Mean	Weekdays Med.
Current Behaviors				
Travel time (hours)	1.23	0.92	1.09	0.87
Riding frequency	2.27	2	1.93	2
Number of trips	1.84	2	1.75	2
Number of short rides	0.45	0	0.29	0
Number of boarding stations	2.24	2	1.91	2
Number of regions	2.08	2	1.84	2
Number of functional transitions (See Section 4.3)	1.97	2	1.77	2
Number of rides on the most frequent route	1.01	1	1.01	1
Maximum number of visits of a functional region	1.19	1	1.09	1
Number of wandering concentration spots (See Section 4.4)	2.99	3	2.61	2
Social Comparisons (See Section 4.5)				
Time gap of trips (hours)	0.48	0.43	0.46	0.40
Time gap of region transitions	0.35	0	0.33	0
Historical Behaviors (See Section 4.6)				
Daily travel time, median	1.11	0.96	1.16	1.03
Daily riding frequency, median	2.17	2	2.24	2
Number of trips, median	1.77	2	1.84	2
Number of short rides, median	0.33	0	0.31	0
Number of boarding stations, median	2.15	2	2.22	2
Number of regions, median	2.01	2	2.08	2
Functional transitions, median	1.91	2	2.01	2
Daily travel time(hour), std. dev.	0.99	0.29	0.95	0.25
Daily riding frequency, std. dev.	0.98	0.62	0.94	0.58
Number of trips, std. dev.	0.49	0.27	0.47	0.28
Number of short rides, std. dev.	0.33	0.20	0.31	0.19
Number of boarding stations, std. dev.	0.92	0.57	0.88	0.55
Number of regions, std. dev.	0.71	0.44	0.68	0.43
Functional transitions, std. dev.	0.89	0.58	0.84	0.54
Number of days detected as suspect	0.00	0	0.00	0

5.2 Regular Passenger Filtering

The first step in our framework is regular passenger filtering using an anomaly detection algorithm. The objective is to exclude regular passengers without any suspicious behaviors from later modeling steps. Therefore, we intentionally allow false-positives in this process. Many general purpose anomaly detection algorithms can be used to implement the filtering function g(·). However, two main challenges must first be addressed. First, pickpockets are a small fraction of the passenger population. Second, the data size is extremely large. There are millions of passenger records amassed daily. To efficiently locate the needles in the haystack, we take advantage of an undersampling-based ensemble learning method. Although undersampling can effectively balance the ratio of the anomalies in the data size, the sampled data might lose a degree of generality since only part of the original data will be used for predictive modeling. To improve the degree of generality with the undersampled and balanced data, we design the undersampling-based ensemble learning method. Specifically, given a base predictive model, we repeat the undersampling of the regular passengers M times, and each time we fit the predictive model and have a regular passenger filtering function gm(·), m = 1, 2, · · · , M. Then, given a new test passenger x, we have the following ensemble filtering function,

$$g(x) = \frac{\sum_{m=1}^{M} w_m \times g_m(x)}{\sum_{m=1}^{M} w_m},$$

where wm is the mixing weight proportional to the expected accuracy of the individual filtering function gm(·). The choice for the base predictive model and the number of base models in the ensemble will be determined based on results from a validation dataset (details in Section 6).

5.3 Suspect Detection

The second step in our framework is suspect detection, which aims to eventually identify the suspected pickpockets. After the regular passengers were filtered in the first step, we were left with positive subjects as declared by the under sampling-based ensemble learning method. This positive group should include most, if not all, real suspects, and possibly many false-positives. However, after the step of regular passenger filtering, the number of the false positives are limited and comparable with the number of suspects. The second step further distinguishes these two subsets with supervised information verified by the social media or security agencies. Specifically, suppose there are Nb total subjects after filtering in the dataset {(xj , yj) | j = 1, · · · , Nb}, where Nb ⍰ N, and yj = 1 if and only if the passenger associated with feature vector xj is a verified pickpocket, otherwise yj = 0. Therefore, to train the suspect detection model, we use the SVM method due to its superior detection accuracy and modeling flexibility [32]. SVM computes non-linear decision/classification boundaries using appropriate kernel functions and soft margins. The kernel function κ(·, ·) is defined as

$$\kappa(x1, x2) = h\varphi(x1), \varphi(x2)i.$$

The function φ(·) maps the original features into a high dimensional kernel space where the optimal decision boundary exists. Also, it can be shown that the optimization process requires κ(·, ·) instead of an explicit formulation of φ(·). In this paper, we use the popular Gaussian kernel:

$$\kappa(x1, x2) = e^{-kx1-x2k\,2/h}. \quad (2)$$

With the decision/classification boundary in the kernel space:

$$h(x) = hw, \varphi(x)i + \rho,$$

we have:

$$h(x) = \begin{cases} 1 & \widehat{h}(x) \geq 0 \\ 0 & \widehat{h}(x) < 0. \end{cases} \quad (3)$$

To compute the optimal w and ρ in $h(\cdot)$, we optimize:

$$\begin{aligned} \min_{w,\rho,\xi} \ & \frac{1}{2}\|w\|^2 + C \cdot \sum_{j=1}^{\widehat{N}} \xi_j \\ \text{s.t. } & \widehat{h}(x_j) \geq +1 - \xi_j, \forall y_j = 1 \\ & \widehat{h}(x_j) \leq -1 + \xi_j, \forall y_j = 0 \\ & \xi_j > 0, \forall j \end{aligned} \quad (4)$$

Here C is a trade-off parameter controlling the softness of the decision boundary. Both C and the kernel parameter h (a.k.a., bandwidth) in Equation (2) will be determined by cross-validation.

6. DETECTING SUSPECTS IN REAL TIME

For real-world implementation, we can straightforwardly estimate the two-step model offline each day. With data newly available everyday, the re-estimated models can provide better identification performance. However, such a naive update procedure is not efficient enough in largescale data sets. To ensure that the system is practical for real-world usage, we adopted a real-time implementation of the regular passenger filtering step (see the shaded box in Fig. 2) with a dynamic ensemble mechanism, as illustrated in Fig. 12.

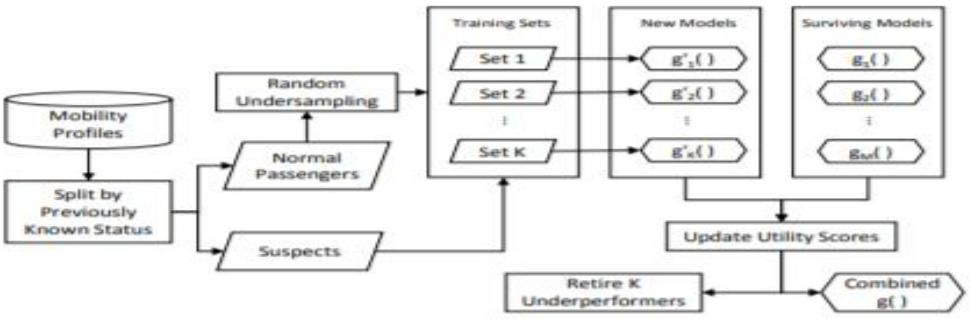


Fig. 12: The incremental update process for training $g(\cdot)$, the regular passenger filtering classifier.

More specifically, we improve the efficiency of regular passenger filtering by maintaining a large number of base filtering models gm(·), for m = 1, 2, · · · , M. Instead of computing these base filtering models independently every day, we dynamically update the ensemble to improve the efficiency. The key idea is to replace the most ineffective or irrelevant base models with new ones, while retaining the majority of the base models from the past. The utility of the model gm(·) is measured by the following utility function:

$$um = \lambda Fm + (1 - \lambda)Rm, \quad (5)$$

which strike for a balance between the effectiveness (Fm) and the relevance (Rm). Since gm(·) is essentially a binary classifier, its effectiveness Fm is measured by a classification performance metric, such as the F-score (see Equation (6)). For relevance, we compute Rm = ln 1 δm+1 , where δm is the "age" of the base model, measured as the number of periods between its creation and now. Intuitively, our relevance definition confers a higher utility to a newer model. Finally, λ ∈ (0, 1) is a parameter balancing the effectiveness and relevance, which can be chosen empirically with respect to the desired accuracy and the available computation capacity. With the utility defined, we dynamically rank the current base filtering models and replace the worst candidates with models estimated with newly available data. Our implementation replaces models with the lowest utility scores.

7. A PROTOTYPE SYSTEM

With the automatic feature extraction and two-step suspect detection model, we developed a decision support system for security personnel to easily spot pickpocket hotspots and apprehend suspects at the crime scenes efficiently. In particular, this prototype system was implemented using bootstrap5 , Java, and SparkQL. Fig. 19 is a screenshot of the graphical user interface (GUI), which can be viewed on a computing terminal or a mobile device. The GUI has the following five basic components, which allow users to view suspect analytics at different levels of detail.

(a) Computing Terminal

(b) Mobile Device

Fig. 19: Screenshots of the prototype system.

• **Statistics**. Summary statistics about the transit system status are provided at the top of the screen, which include the total numbers of passengers, commuters, outliers, and suspects, respectively. Through various settings, the user is allowed to specify the time window for these statistics in terms of the number of days.

• **Passenger Flows.** The density of passenger flows has a high correlation with pickpocket activities. The live state of passenger flow is shown with a heat map, where the density of passenger flow of each station is expressed by blending the color between green and red. (Redder lines indicate higher densities.) This map visually identifies higher trafficked areas that would be more vulnerable to theft

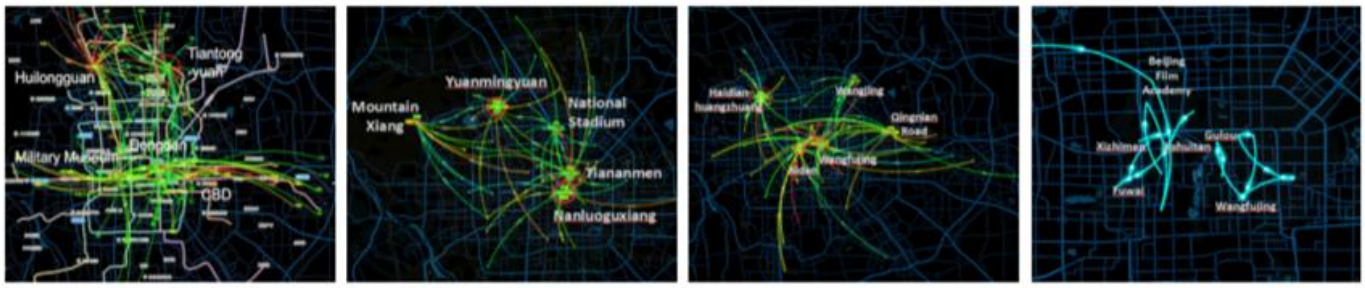
. • **Active Regions**. Active regions of suspects at the city level is visualized in the "active regions" map. Indicated by blue flashing circles, these regions are found by calculating the centroids of a DBSCAN algorithm. By zooming in, the user can inspect a specific area.

• **Suspect List**. Passengers identified as suspects will be listed on the "suspect list." Profiles of these suspects, such as smart card ID number, total travel time, riding frequency, the number of trips, and the number of short rides, will be displayed by default. Quantile score against the social norm, historical profile information, and system determined likelihood of a suspect, are also available. The user is allowed to choose which features to display and to sort.

• **Selected Suspect**. When a suspect on the list is selected, his or her trajectory, as represented by linking the boarding and exiting stations, can be displayed in the "selected suspect" panel. The database is updated every day with newly collected transactions data and identified suspect records. The suspect detection model is pre-computed daily, so detectives can obtain instant result when they interact with the system.

7.1 Typical Passenger Behaviors

As mentioned in the "passenger flow" function above, we visualize the passenger movement patterns on the city map to analyze the behaviors of different types of passengers. Fig. 20 provides examples of representative movement patterns in different passenger groups in Beijing on a typical day from 8:00 a.m. to 11:00 a.m. Each curve in the figure represents the transition between a pair of origin and destination regions, and the color represents the traffic density (red=high, green=low). The overall passenger flow is shown in Fig. 20(a), which provides a bird's-eye view of the most dense traffic at the city level. We can see that the Huilongguan area, Tiantongyuan area, Military Museum, CBD area, and the Dongdan area have the highest densities. We can observe several other gathering regions, such as the Wudaokou area, Olympic Park area, and Beijing West Railway Station. Since passenger flows are mixed, certain special patterns are hard to discover, especially for travel anomalies. After applying our method, such patterns, which typically present remarkably distinctive features, can be revealed. For example, we could classify passengers by major categories of functional regions they visit. While most travelers present commuter patterns like residence-workplace sequences (61.2%), Figures 20(b), 20(c), and 20(d) show typical visitor passenger flow (3.2%), shopper passenger flow (3.8%), and thief passenger flow (0.03%), respectively. Since thieves comprise a tiny percentage of all passenger, they are almost always treated as anomalous data and are neglected. It is also worth noting that visitors patterns resemble thieve patterns in that they both visit many different places. However, visitors are more likely to take longer trips among scenic spots, which are captured in our framework to differentiate them from thieves. Fig. 20(b) shows that visitors frequently visit Yuanmingyuan, Tiananmen, and Nanluoguxiang, whereas Fig. 20(c) shows that shoppers tend to visit regions like Wangfujing and Xidan. Most normal travelers have clear directions, but as visualized in Fig. 20(d), pickpockets tend to wander randomly and make frequent stops without clear destinations. Thieves also tend to visit a mixture of functional regions, such as transit hubs (e.g Xizhimen), shopping regions (e.g., Wangfujing), and scenic spots (e.g., Gulou), whereas most regular passengers only visit one functional region during a short period of time.



(a) all passengers (b) visitors (c) shoppers (d) thieves

Fig. 20: Movement patterns of different type of passengers.

8. EXPERIMENTAL RESULTS

In this section, we present experimental results employing our proposed framework. First, we describe the experimental environments and provide implementation details. We then demonstrate the effectiveness of our framework by comparing it to several baseline methods.

8.1 Experiment Settings

In this subsection, we will outline our experimental environments and study design. This includes a short description on the platform, the baseline methods, and the performance metrics.

TABLE 3: A performance comparison.

	Offline				Online				
Algorithm	Precision	Recall	F-score	Run Time(s)	Precision	Recall	F-score	Init. Time (s)	Update Time (s)
Classification Methods									
DT	0.002	0.451	0.004	44.81	0.017	1.000	0.034	165.86	15.52
LR	0.003	0.476	0.006	36.72	0.067	0.931	0.124	1643.46	56.17
SVM	0.005	0.512	0.009	21.31	0.049	0.931	0.093	2384.15	62.34
Anomaly Detection Methods									
LOF	0.004	0.560	0.009	300.01+	0.037	0.746	0.071	9821.25+	516.54+
OCSVM	0.015	0.583	0.029	39.67	0.099	0.931	0.179	1845.68	45.56
Two-Step Methods									
LOF+DT	0.011	0.780	0.022	301.18+	0.087	0.795	0.157	9821.25+	428.87+
LOF+LR	0.016	0.829	0.031	301.16+	0.093	0.871	0.168	9821.25+	358.79+
LOF+SVM	0.027	0.758	0.052	318.16+	0.097	0.926	0.175	9821.25+	483.57+
OCSVM+DT	0.053	0.878	0.099	41.19	0.065	0.754	0.120	987.23	63.28
SVM+LR	0.059	0.855	0.110	85.43	0.047	0.931	0.090	2384.15	112.12
OCSVM+SVM	0.071	0.927	0.133	41.05	0.097	0.891	0.175	987.23	65.74
LR+SVM	**0.117**	**0.931**	**0.207**	**65.72**	**0.169**	**0.931**	**0.288**	**1643.46**	**74.35**
LR+DT	0.093	0.985	0.170	45.87	0.120	1.000	0.214	1643.46	71.21
DT+SVM	0.086	0.925	0.157	37.54	0.114	0.931	0.203	1274.69	64.23
SMOTE/SVM+LR	0.043	0.845	0.082	71.69	0.041	0.845	0.078	1984.65	112.12
SMOTE/LR+SVM	0.103	0.931	0.185	60.57	0.135	0.754	0.228	1453.54	74.35
SMOTE/LR+DT	0.074	0.952	0.137	43.87	0.094	0.931	0.171	987.32	71.21
SMOTE/DT+SVM	0.082	0.913	0.150	35.45	0.085	0.891	0.155	985.54	64.23

Platform. All offline experiments were conducted on a Windows Server 2012 64-bit system (4-CPU, each with 2.6GHz with Quad-Core, and 128G main memory). The realtime system was implemented on a Spark cluster with 10 nodes. Each node has a Intel i7-4790 CPU3.6GHz CPU with 8 cores, 2*8GB Kingston Memory, 2 TB SATA3.0 hard drive, with a Centos 6.5 Operation System. All algorithms and our real-world system were developed with Java and Scala.

Data Preparation. All experiments were conducted on real-world datasets described in Section 3. There were about 1.7 billion records collected between April and June in 2014. We eliminated passengers from the training set whose maximum number of daily records is no more than three. After removing duplicates and extremely infrequent riders, we had over 1.6 billion records remaining that involve approximately 6 million passengers over the three-month period. We split the data into a training set for model building and a testing set for evaluations. Specifically, for offline experiments, the training set covers three months (from April to June, 2014) and the test set derives from the following two weeks (in July 2014). For evaluating the real-time system, we let the models train incrementally and reuse data over time. Each day, there are about 14 million records collected from around 5 million individuals.

Baselines. Our method is compared with a variety of competing methods grouped into the following categories: • Classification Methods. The classification methods, including logistic regression (LR), decision trees (DT) [33], and SVM [32], are straightforwardly fit to the training set and evaluated with the test set. Since the proportion of positive instances is extremely low, the classification problem is unbalanced, and we expected to observe high Type II Error. In the experiments, we under-sample the negative instances to balance the data and improve the results. For each method, we repeat the sampling 10 times and report the averaged results.

• **Anomaly Detection**. Anomaly detection methods, such as one-class SVM (OC-SVM) [34] and local outlier factor (LOF) [20], seem more appropriate for our problem. Among them, LOF is unsupervised, finding outliers by measuring the local deviation of a given data point with respect to its neighbors. OC-SVM can be fitted in a supervised manner, with only the negative instances in the training set, to identify the suspects.

• **Two-Step Methods.** As previously mentioned, our approach is a TS method, consisting of a negative sample filtering and then applying a traditional classification step. For specific combination of techniques, we experimented a number of possibilities. For all of the methods with parameters, we optimized the parameters with 10-fold cross-validation by further dividing the training set with 80% for model fitting and 20% for parameter validation. We separately evaluated the methods in both the offline mode and an online mode. In addition, to demonstrate the power of dynamic ensemble in the regular passenger filtering step, we compared it with an alternative approach to create balanced training sets using SMOTE [35]. Evaluation Metrics. Precision, recall, and the F-score were computed based on the test set to evaluate the performances of different methods. Precision is the number of correctly identified positives divided by the number of identified positives instances. Recall is the number of correctly identified positives divided by the number of all positive instances in the test set. And finally, the F-score is calculated as

$$\text{F-score} = 2 \times \frac{precision \times recall}{precision + recall}. \qquad (6)$$

8.2 Modeling Performance

Table 3 summarizes the performances of our method (with various two-step settings) and the baselines in both offline and real-time systems. In the offline mode, we arrived at several interesting observations from the modeling performance. First, the precision of one-step methods is generally low. In contrast, all the two-step combinations significantly improve the precision, with the OCSVM-SVM setup exhibiting the best performance. This observation shows that the two-step approach can effectively reduce the false-positives. Second, two-step methods also performed better in terms of recall and the F-score. Finally, the precision of all methods are low. This was intentional since we wanted to ensure a high recall. After all, our ground truth (i.e., flagged suspects) only consists of those confirmed pickpockets, whereas it is likely there are many more pickpockets that go unapprehended. With

an dynamic ensemble in the real-time implementation, while maintaining a high recall, the two-step model with LR+SVM specification performed the best, resulting in the highest F-score. For the real-time system, we separated two types of running time: the initialization time and the update time, both measured in seconds. The initialization time is relatively long, given that we need to train an ensemble of many base models. The LR+SVM model performs well overall in terms of running time.

8.3 Parameter Tuning

One of the most important parameters we needed to decide was the number of base classifiers to employ. Considering that the behavior of one person may change over time, occasionally even be abnormal, we needed to use a long period of transit records to extract features and train classifiers.

In Fig. 13, we show the trend in performance as we increase the number of base classifiers. The graph illustrates that both the precision and the F-score changes as the number of base classifiers M grows from a smaller number to 241. When M grows beyond 241, the performance metrics stay level or start to decrease. Therefore, we chose M = 241 as the number of combination classifiers.

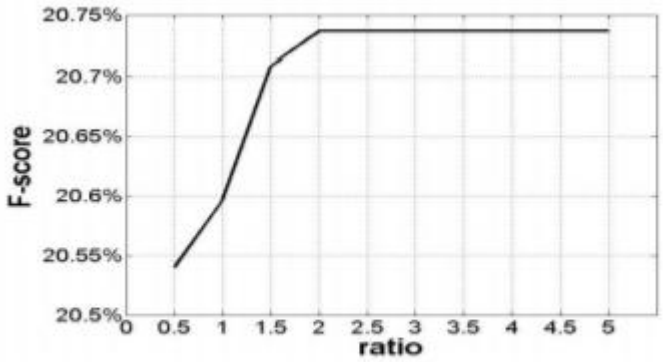


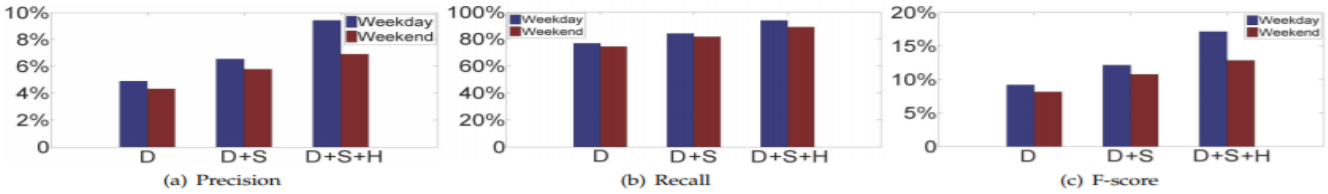
Fig. 14: Parameter selection for the utility function

According to the utility definition in Equation 5, the utility value depends on the relative importance of the F-score and the recall by the ratio of λ $1-\lambda$. After experimenting with a number of possible values for this tradeoff parameter, as shown in Fig. 14, we can see that the modeling performance (i.e., F-score) reaches its peak when the ratio is at least 2 (i.e., when $\lambda = 0.67$).

8.4 Feature Analysis

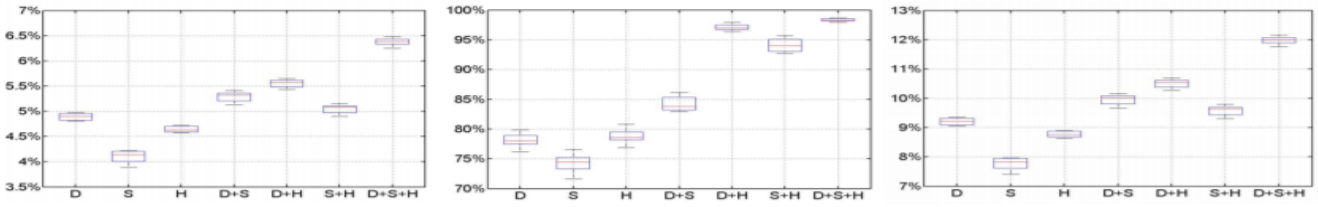
To further study the discriminative power of the features, we evaluate the performance of our framework with different feature combinations. As shown in Fig. 15, we use D, S, and H to represent the daily behavior, social comparison, and historical behavior features, respectively. Most significantly, the precision of the daily behavior features is improved by the social comparison, and further by the historical behaviors. Such improvements can also be observed for metrics shown in the other subfigures. In Fig. 15, we also compared the modeling performances on weekdays and weekends. As expected, since human mobilities during weekends are more complicated, the detection accuracy of our method was slightly lower on weekends. For the real-time system, we evaluated the contribution of various combinations of features in a similar way. The result is shown in Fig. 16, which demonstrates that combining features helped improve the model performance. In particular, combining all three types of features led to the best performance in our experiments. 7.5 Example Trajectories In this section, we provide a few example trajectories (in blue lines) projected on the map for qualitative evaluation. To ensure individuals' privacy, we chose densely populated areas so that no one was uniquely identifiable. We also placed the

actual location points to nearby points whenever we could to blur the actual locations visited. Each numbered box in a figure represents a station where the individual left a vehicle or boarded another one. The numbers in the boxes are sequence IDs. An example of a confirmed thief's one-day trajectory is shown in Fig. 17. With all activities shown, we could clearly see that Pingguoyuan and Shijingshan subway stations



(a) Precision (b) Recall (c) F-score

Fig. 15: The contribution of feature combinations for weekdays and weekends.



(a) Precision (b) Recall (c) F-score

Fig. 16: The contribution of feature combinations for the real-time system.

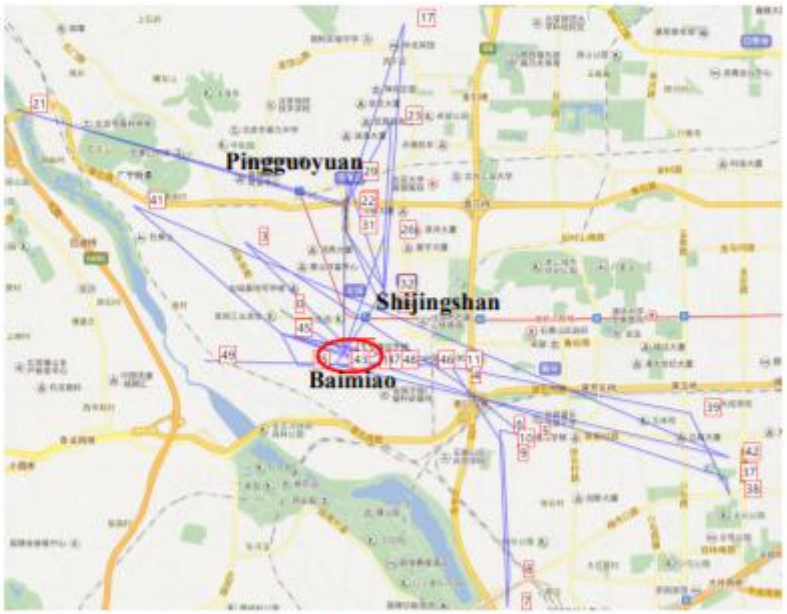


Fig. 17: The trajectory of a thief.

(a) Suspect (b) Visitor

Fig. 18: Example location traces.

were gathering regions for this particular thief. This was an important discovery, as it told detectives which regions were hubs for thieves and how many suspects were active in that area. To know who is the next target, it is necessary to first provide a track of a thief in the active areas in order to better enable detectives' ability to catch thieves at the crime scene. Fig. 17 also reveals strong mobility patterns in thief activity. For instance, we see that Baimiao station, a hub transfer subway station north of Beijing, is considered a center by the thief, rather than a route to actively engage in pickpocketing. Although the thief will attempt to elude authorities by traveling to neighboring stations, the key finding is that he/she will return to the station eventually. To intuitively demonstrate the rationale of our approach, we provide a specific case study in Fig. 18. This figure shows the trajectories of two passengers during the Cherry Blossom Season of 2014 in Beijing. Fig. 18(a) shows the traces of one identified, confirmed pickpocket A, while Fig. 18(b) shows the traces of one Cherry Blossom tourist B. Both of these passengers are quite unusual in comparison with daily commuters, but our model can successfully distinguish between them based on the comprehensive features previous defined. In particular, the features by social comparison of A and B are quite different because there were many other Cherry Blossom visitors following the traces of B on that same day. In addition, the historical behavior of B is more in keeping with his/her home-work routines.

9. CONCLUSION

In this paper, we developed a suspect detection and tracking system by mining large-scale transit records. The system assists in identifying pickpocket suspects' and enables active surveillance in high-risk areas. Specifically, we first constructed a feature representation for profiling passengers. Then, we established a novel two-step framework to distinguish regular passengers from pickpocket suspects. Finally, we leveraged real-world datasets from multiple sources for model training and validation, and implemented a prototype system for end users. Experimental results on real-world data showed the effectiveness of our proposed approach.

ACKNOWLEDGMENTS

This research was supported in part by National Natural Science Foundation of China (No. 51408018, No. 51778033, and No. 71329201), National High Technology Research and Development Program (863, 2013AA01A601). The authors would like to thank the anonymous reviewers and KDD 2016 conference participants for their helpful comments and constructive feedback.

REFERENCES

[1] G. R. Newman and M. M. McNally, "Identity theft literature review," United States Department of Justice, Report 210459, July 2005.

[2] M. Felson and R. V. Clarke, "Opportunity makes the thief: Practical theory for crime prevention," Policing and Reducing Crime Unit: Police Research Series, Report 98, 1998.

[3] T. L. C. da Silva, J. A. F. de Macedo, and M. A. ˆ Casanova, "Discovering frequent mobility patterns on moving object data," in MobiGIS, 2014, pp. 60–67.

[4] I. Ceapa, C. Smith, and L. Capra, "Avoiding the crowds: understanding tube station congestion patterns from trip data," in UrbComp, 2012, pp. 134–141. [5] X. Ma, Y.-J. Wu, Y. Wang, F. Chen, and J. Liu, "Mining smart card data for transit riders travel patterns," Transportation Research Part C, vol. 36, pp. 1–12, 2013.

[6] K. Zheng, Y. Zheng, N. J. Yuan, S. Shang, and X. Zhou, "Online discovery of gathering patterns over trajectories," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 8, pp. 1974–1988, 2014.

[7] Y. Liu, C. Liu, J. Yuan, L. Duan, Y. Fu, H. Xiong, S. Xu, and J. Wu, "Intelligent bus routing with heterogeneous human mobility patterns," Knowledge and Information Systems, 2016, forthcoming, DOI: 10.1007/s10115-016- 0948-6.

[8] B. Du, C. Liu, W. Zhou, Z. Hou, and H. Xiong, "Catch me if you can: Detecting pickpocket suspects from large-scale transit records," in KDD, 2016, pp. 87–96.

[9] C. Morency, M. Trepanier, and B. Agard, "Analysing ´ the variability of transit users behaviour with smart card data," in ITSC, 2006, pp. 44–49. [10] M. J. Sung, "Analysis of travel patterns of the elderly using transit smart card data," in TRB, no. 11-2357, 2011.

[11] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, ´ "Understanding individual human mobility patterns," Nature, vol. 453, no. 7196, pp. 779–782, 2008.

[12] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi, "Limits ´ of predictability in human mobility," Science, vol. 327, no. 5968, pp. 1018–1021, 2010.

[13] C. Chen, D. Zhang, Z.-H. Zhou, N. Li, T. Atmaca, and S. Li, "B-planner: Night bus route planning using largescale taxi gps traces," in PerCom, 2013, pp. 225–233.

[14] W. Luo, H. Tan, L. Chen, and L. M. Ni, "Finding time period-based most frequent path in big trajectory data," in SIGMOD, 2013, pp. 713–724.

[15] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in KDD, 2012, pp. 186–194.

[16] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," in KDD, 2011, pp. 1010–1018.

[17] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," in GIS, 2013, pp. 344–353.

[18] L. Hong, Y. Zheng, D. Yung, J. Shang, and L. Zou, "Detecting urban black holes based on human mobility data," in GIS, 2015, pp. 35:1–35:10.

[19] P. Bouman, E. Van der Hurk, L. Kroon, T. Li, and P. Vervest, "Detecting activity patterns from smart card data," in BNAIC, 2013. [20] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," SIG-