Detecting Vehicle Illegal Parking Events using Sharing Bikes' Trajectories

Tianfu He[1], Jie Bao[2], Ruiyuan Li[3,2], Sijie Ruan[3,2], Yanhua Li[4], Chao Tian[5], Yu Zheng[2,3]
[1]Harbin Institute of Technology
[2]Urban Computing Business Unit, JD Finance
[3]School of Computer Science and Technology, Xidian University, China
[4]Worcester Polytechnic Institute, USA
[5]Beijing Mobike Technology Co., Ltd.
hetianfu@hit.edu.cn;{baojie,ruiyuan.li,ruansijie}@jd.com
yli15@wpi.edu;tianchao@mobike.com;msyuzheng@outlook.com

ABSTRACT

Illegal vehicle parking is a common urban problem faced by major cities in the world, as it incurs traffic jams, which lead to air pollution and traffic accidents. Traditional approaches to detect illegal parking events rely highly on active human efforts. However, these approaches are extremely ineffective to cover a large city.

The massive and high quality sharing bike trajectories from Mobike offer us with a unique opportunity to design a ubiquitous illegal parking detection system, as most of the illegal parking events happen at curbsides and have significant impact on the bike users. Two main components are employed in the proposed illegal parking detection system: 1) *trajectory pre-processing*, which filters outlier GPS points, performs map-matching and builds trajectory indexes; and 2) *illegal parking detection*, which models the normal trajectories, extracts features from the evaluation trajectories and utilizes a distribution test-based method to discover the illegal parking events. The system is deployed on the cloud, and used by Mobike internally. Finally, extensive experiments and many insightful case studies are presented.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**;

KEYWORDS

Trajectory Data Mining, Urban Planning, Urban Computing

ACM Reference Format:
Tianfu He[1], Jie Bao[2], Ruiyuan Li[3,2], Sijie Ruan[3,2], Yanhua Li[4], Chao Tian[5], Yu Zheng[2,3]. 2018. Detecting Vehicle Illegal Parking Events using Sharing Bikes' Trajectories. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219887

Jie Bao is the corresponding author.

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08.
https://doi.org/10.1145/3219819.3219887

(a) Illegal Parkings (b) Traditional Illegal Parking Detection Methods

Figure 1: Issues with Illegal Parking.

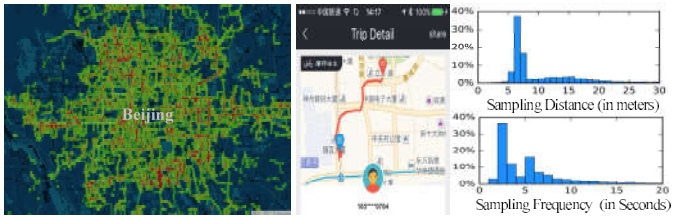
1 INTRODUCTION

Illegal vehicle parking is a common problem in large cities all over the world. The illegal parking events decrease the transportation efficiency in a city, and incurs traffic jams [21], which lead to air pollution [13] and potential accidents (as illustrated in Figure 1a, bike users have to ride on the vehicle lanes). Effective detection of illegal vehicle parking events can improve the effectiveness of city management (e.g., planning more effective police patrol routes) and urban planning (e.g., building more parking spaces in the illegal parking hotspots) for the government.

Traditional approaches to detect illegal vehicle parking events rely mainly on human efforts, e.g., police patrols, where the illegal parking events are detected only if they are in the sight (e.g., Figure 1b). With the advances of video object identification technologies, different algorithms emerge, e.g., [14, 27], to identify the illegal parking events based on surveillance cameras (as Figure 1b). However, all of the existing approaches (police patrols and surveillance cameras) are active detection methods, and can only cover limited spatial ranges, which makes them highly ineffective and costly, to achieve a high coverage level in large cities.

To this end, in this paper, we propose a ubiquitous approach to effectively detect illegal vehicle parking events by mining the trajectories of sharing bikes. The intuition behind the technique is that, based on our observation, illegal vehicle parking events usually take place at curb sides, which block the path of bike users and significantly affect their trajectories. Therefore, by aggregating massive bike trajectories on the same road, we are able to identify the illegal parking events via examining the distinct patterns of their trajectories. Fortunately, the bike trajectory data provided by Mobike [1] (a station-less sharing bike service provider in China),

[1]https://en.wikipedia.org/wiki/Mobike



(a) Wide Usage Coverage (b) Detailed Trajectory (c) Fine Data Granularity

Figure 2: Opportunity in Mobike Trajectories.

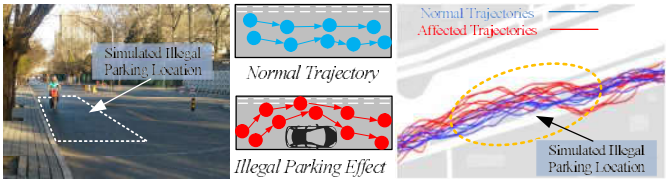
offers us a unique opportunity to tackle the problem with two distinctive advantages:

- **Wide Usage Coverage.** Mobike is a very popular bike sharing service, which is frequently used as a daily commute mode for many people nowadays. According to the recent report [2], it got more than 200 million registered users and 30 million daily trips. Moreover, Mobike trajectories cover widely across a city, e.g., Figure 2a visualizes the Mobike usages in the city of Beijing. With the most roads in the urban area densely covered, it is possible for us to detect illegal parking events in large cities without any active efforts.
- **High Data Quality.** First of all, Mobike records detailed GPS trajectories for each trip, as demonstrated in Figure 2b. Moreover, the granularity of each trajectory is very high, as shown in Figure 2c: 1) more than 60% of the distances between two GPS points are less than 6 meters, and 2) over 70% of the time interval between two GPS points are less than 6 seconds. Therefore, it is possible for us to identify the subtle travelling behaviour changes, caused by vehicle illegal parking events.

With the access to the large-scale and high-quality Mobike trajectories, we first conduct a set of experiments to validate the feasibility of our intuition, i.e., whether it is possible to identify illegal parking events based on sharing bike trajectories. We ride a Mobike on a local road (as shown Figure 3a multiple times, where the area marked in the white lines is the simulated illegal parking location), with two settings: 1) with simulated illegal parking vehicles, and 2) without simulated illegal parking vehicles, each for ten times, i.e., conceptually demonstrated in Figure 3b.

Figure 3c visualizes the experimental trajectories extracted from Mobike, where the red lines are the trajectories with illegal parking simulation, and the blue lines are the normal trajectories. It is clear that, especially around the simulated illegal parking location (marked with the orange circle), comparing to the normal trajectories, the affected trajectories are more twisted and leaning toward the opposite side of the curbside. As a result, this set of experiments confirms our intuition.

However, to realize this idea, there are still many challenges: 1) *data errors*, caused either by the GPS module or human errors (e.g., forget to lock and return bike); 2) *map-matching*, which is a crucial step to match trajectories to correct road segments. It is more difficult working with bike trajectories, as bikes can be ridden more freely beyond the roads; 3) *illegal parking detection*, developing an effective detection model is not trivial, as it is hard to collect massive labelled data; and 4) *system efficiency*, with the massive trajectories in a large city, the system response time needs



(a) On-Field Scenario (b) Experiment Method (c) Results

Figure 3: Intuition Validation Experiments.

to be minimized for users (i.e., city managers) to identify all illegal parking events in a city.

In this paper, we design, implement and deploy an illegal parking detection system based on data mining results from the massive sharing bikes' trajectories. The system consists of two main modules: 1) *pre-processing*, which filters outlier GPS points, performs map-matching, and builds indexes; and 2) *illegal parking detection*, which studies a baseline to model the normal trajectories for each road, extracts the features of evaluation trajectories and infers the possibility of the presence of illegal parking events. To improve the system response time and efficiency, the trajectory data is stored on a distributed storage platform, i.e., MongoDB and the illegal parking detection system is deployed on Apache Storm. The main contributions of the paper are summarized as follows:

- We provide the first attempt on detecting illegal parking events ubiquitously by mining massive bike trajectories.
- We design and implement a comprehensive *pre-processing* module to clean bike trajectories, map them to corresponding road segments and build a set of indexes. We also propose a novel distribution test-based approach to detect the illegal parking events on a road segment.
- We collected over 400 illegal parking labels manually to tune the most effective threshold in the detection model.
- We evaluate the proposed model extensively over six months' Mobike trajectory data from the City of Beijing. Moreover, on-site case studies are conducted to validate the effectiveness of our illegal parking detections.
- A system is deployed on the cloud and used internally [1].

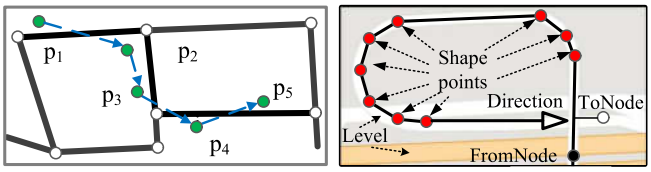
The rest of the paper is organized as follows: Section 2 describes the problem and the system overview. Section 3 presents the preprocessing module. Illegal parking detection is discussed in Section 4. Section 5 presents the interface and deployment. Experiments and case studies are given in Section 6. Related works are summarized in Section 7. Section 8 introduces the future work. Finally, Section 9 concludes the paper.

2 OVERVIEW

In this section, we first present the preliminary concepts. After that, we formulate the illegal parking detection problem. Finally, an overview of the our proposed system is demonstrated.

2.1 Preliminaries

Definition 1. ***(Trajectory)*** *A trajectory τ can be defined as a time-ordered sequence $\tau = \{p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_n\}$, where $p_i =$*



(a) Trajectories & Road Networks (b) Road Segment Properties

Figure 4: Examples of Preliminary Concepts.

$(lat_i, lng_i, t_i), 1 \le i \le n$, *is a GPS record with latitude* lat_i, *longitude* lng_i, *and timestamp* t_i.

The Mobike trajectory starts when the user scans the QR code to unlock a sharing bike, and ends when the user locks the bike. The intermediate GPS points are sampled at a constant rate.

Definition 2. ***(Road Network)*** *A road network RN is a directed graph* $G = (V, E)$, *where* $V = \{v_1, v_2, ..., v_m\}$ *is a set of intersections, and* $E = \{e_1, e_2, ..., e_n\}$ *is a set of road segments (edges).*

For each $e_i \in E$, *it associates with three properties: 1)* level, *which indicates the type of road; 2)* shape, *which is a sequence of location points, from a* FromNode *to a* ToNode, *describing the shape of the segment; and 2)* dir, *which indicates its directional information (bidirectional or uni-directional).*

Figure 4a gives an example of the concepts. The green dots are GPS points, and the blue arrows indicate their sequence. On the other hand, the white dots are the nodes and lines are the edges of the road network. Figure 4b illustrates the detailed properties of a road segment, where the red dots are the location points describing the *shape*, the white dot and the black dot are the FromNode and ToNode respectively, the arrow indicates its *dir* property (unidirectional in this case), and the colors represent different levels of roads, e.g., the yellow color represents highways, and the white color means supplementary roads.

Definition 3. ***(Illegal Parking Event)*** *An illegal parking event in this paper refers to obstacles at a road segment* e_i *that affects the normal behavior of bike trajectories* (Tr_{e_i}) *on it, during a temporal range* t_i & t_{i+1} *(e.g., 8:00 AM to 9:00 AM).*

2.2 Problem Definition

We now formalize our illegal parking detection problem as follows.

Given a set of trajectories Tr, *a road network* $G = (V, E)$, *and temporal period* t_i & t_{i+1}, *for every road segment (or edge* $e_i \in E$),

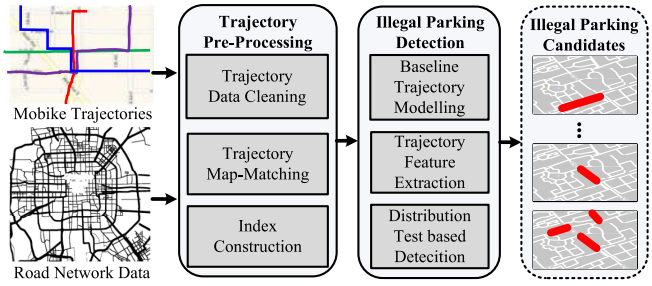
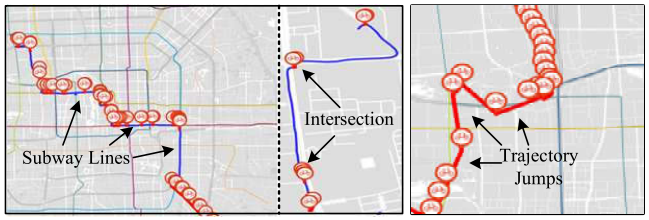


Figure 5: System Overview.



(a) Abnormal Trip Speed (b) Low Sampling Rate

Figure 6: Examples of Error Data.

we want to infer the possibility of the presence of the illegal parking events, based on the sharing bikes' trajectories generated on each road segment e_i *from* t_i & t_{i+1}. *The objective is to achieve a high accuracy of the detection, as well as reduce the system response time.*

2.3 System Overview

Figure 5 gives a system overview with two main components:

Pre-Processing. This component takes bike trajectories and road networks and performs three main tasks: 1) *Trajectory Data Cleaning*, which removes the outlier GPS points; 2) *Trajectory Map-Matching*, which projects GPS points onto the corresponding road segments; and 3) *Index Construction*, which builds indexes to speed up the trajectory retrieval process based on road segment IDs and temporal ranges (detailed in Section 3).

Illegal Parking Detection. This component calculates a score for each road segment, indicating the probability of the presence of illegal parking events, by evaluating the processed trajectories in a temporal period. Three main tasks are performed: 1) *baseline trajectory modelling*, which builds a model for each road segment to describe the normal trajectories; 2) *trajectory feature extraction*, which extracts the features from the evaluation trajectories; and 3) *distribution test-based detection*, which detects illegal parking events using distribution tests (detailed in Section 4).

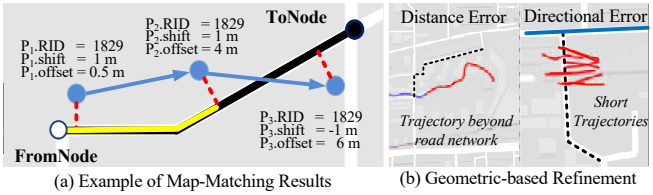
3 TRAJECTORY PRE-PROCESSING

As the data quality of bike trajectories used in our system determines the accuracy of the illegal parking detection, a set of pre-processing tasks are necessary, before the massive trajectories from Mobike users can be used: 1) *Trajectory Data Cleaning*, which removes the GPS outliers in a trajectory based on the speed and sampling rates; 2) *Trajectory Map-Matching*, which segments the GPS points in the trajectories and maps them onto the corresponding road segments; and 3) *Index Construction*, which builds the indexes to speed up the trajectory data retrieval process.

3.1 Trajectory Data Cleaning

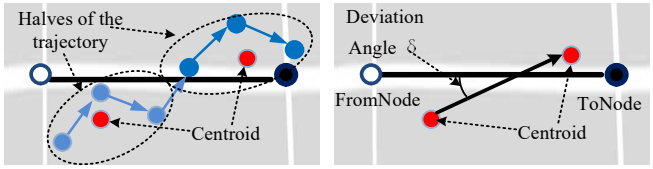
This module cleans the raw trajectories from Mobike. Essentially as a type of crowd sensing data, Mobike trajectories are generated by the GPS modules from mobile phones. As a result, a noticeable portion of trajectories have different data errors, which significantly affect the accuracy of illegal parking detection:

(1) **Abnormal Speeds.** As most users ride bikes at normal speeds (e.g., 5 kmph to 20 kmph), there are two types of abnormal speeds: 1) *abnormal high speed*, which is caused by



(a) Example of Map-Matching Results (b) Geometric-based Refinement

Figure 7: Trajectory Map-Matching.



(a) Centroid Nodes Calculation (b) Deviation Angle Calculation

Figure 8: Map-Matching Refinement.

GPS errors, or unusual usage (e.g., demonstrated as the left portion of Figure 6a, where a user got on a subway without locking the bike); and 2) *abnormal low speed*, which is usually caused by the traffic lights (demonstrated as the right portion of Figure 6a).

(2) **Low Sampling Rates.** In some cases, due to the errors of GPS modules in users' mobile phones, some of the GPS points may be missing. Figure 6b shows an example of a bike trajectory travelling with a normal speed, but with several jumps (marked in red lines).

Both of the above data quality issues introduce problems in the detection model. For example, the trajectory segments with abnormal low speed can be affected by many factors other than illegal parkings, e.g., pedestrians. Moreover, the trajectory segments with low sampling rates introduce challenges in map-matching and provide no information reflecting the conditions of the roads. As a result, a heuristic based approach [38] is used here to clean the trajectory data: If any portion of the consecutive GPS points fails to meet the speed or sampling rate thresholds, these disqualified segments are removed from the original trajectory. In the end, one trajectory may be segmented into several short qualified sub-trajectories to preserve more information.

3.2 Trajectory Map-Matching

In this module, we map the GPS points onto the corresponding segments in road networks, which is crucial for the illegal parking detection. Traditional map-matching algorithms, e.g., [37], cannot be used directly, because, comparing to vehicle trajectories, bike trajectories have several unique properties: 1) they have much lower travelling speeds, 2) they travel at both directions even at a uni-directional road, 3) they can go to the area without road networks; and 4) they have more short trips. To adapt with these properties, the map-matching module is designed with three steps:

Step 1. Adaptive Map-Matching. This step employs an interactive-voting based map matching algorithm [37] with three modifications: 1) high level roads, which can be only used by vehicles (e.g., highways), are removed; 2) the direction information on road segments is omitted, and all road segments are set as bidirectional; and 3) the speed constraint of each road segment is not used to adapt the slower speed in bike trajectories.

After the map-matching process, as shown in Figure 7a, each GPS point is associated with three new properties: 1) *RID*, which is the map-matched road segment ID; 2) *shift*, which is the shortest distance to the road segment, illustrated as the red dotted lines. We define the positive shift for the GPS points at the left side of a road segment (direction is from `FromNode` to `ToNode`, as P_1&P_2), and negative shift at the right side of the road (as P_3); and 3) *offset*, which is the length between the `FromNode` and the projection of the GPS point, as the yellow segment is the offset for P_2.

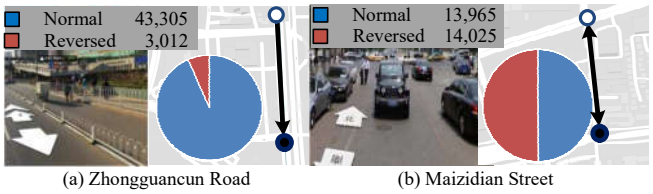
Step 2. Geometric-based Refinement. This step removes the problematic map-matching results, using geometric filters. Figure 7b shows two types of problematic map-matching results: 1) *d*istance error, which is caused by the incompleteness of the road network data. As demonstrated in the left portion of Figure 7b, the algorithm maps the trajectory inside a residential area onto the black dotted road segments around it. It is because the road network we used is not detailed enough to reflect these small roads; and 2) *directional error*, which is caused by the short trips in the data set, generated by the users or as the result of the trajectory data cleaning process. The right portion of Figure 7b shows many short trips (marked in red), which are mapped onto the black dotted road segment, while it makes more sense to map them onto the blue segment, as they head similar directions.

We use a geometric-based refinement to remove these errors, with the consideration of random shifts incurred by the GPS sensors. First of all, the distance errors are removed, if the average shift a sub-trajectory is greater than a threshold (e.g., 20 meters in our implementation). To remove the directional errors, a deviation angle is calculated between the directions of the overall trajectory and the road, as demonstrated in Figure 8. The overall trajectory direction is calculated by connecting the centroid points (i.e., the red dots) between the first and second portion of the sub-trajectory (i.e., circled by the dotted ovals). The trajectory is removed, if the deviation angle δ is greater than $\frac{\pi}{3}$.

Step 3. Reverse Trajectory Removal. This step removes the trajectories travelling at the reverse direction of the uni-directional roads. As all the roads are considered as bidirectional in the map-matching step, for a uni-directional road, there are a small number of reverse travelling trajectories by the users disobeying the traffic rules. Although the number of reverse travelling trajectories is limited, they usually have higher shift values, as they are much likely to encounter obstacles other than illegal parking events, e.g., bikes travelling at the normal direction. Therefore, the reverse trajectories affect the accuracy in our illegal parking detection model (experiments are provided).

To identify the reverse travelling behaviours, we calculate the overall direction of a trajectory by comparing the average offset between the two halves of the trajectories. On a uni-direction road, if the average offset of the first half is less than it is in the second half, the trajectory travels reversely on the road segment. Figure 9 shows the distribution of normal and reversed trajectories we identified on a uni-direction road, i.e., Zhongguancun Road and a bi-directional road, i.e., Maizidian Street. The reverse trajectory

identification result is consistent with our intuition, where much less numbers of reverse trajectories (less than 10%) appear in a uni-direction road. On the other hand, the distribution is more balanced (i.e., 50% for each direction) on the bidirectional road. Finally, all the reverse trajectories are removed.



(a) Zhongguancun Road (b) Maizidian Street

Figure 9: Results of Reverse Trajectory Identification.

3.3 Index Construction

In this module, the system builds an *inverted index* based on each directional road segment (i.e., two entries are built for a bidirectional road segment), where each entry associated with the trajectories are mapped to it. Moreover, a temporal index is built based on the time stamp when the trajectory enters the road segment, as most of the latter trajectory retrieval tasks are based on the road segment ID and a temporal range. In our implementation, all trajectories and indexes are maintained in a MongoDB.

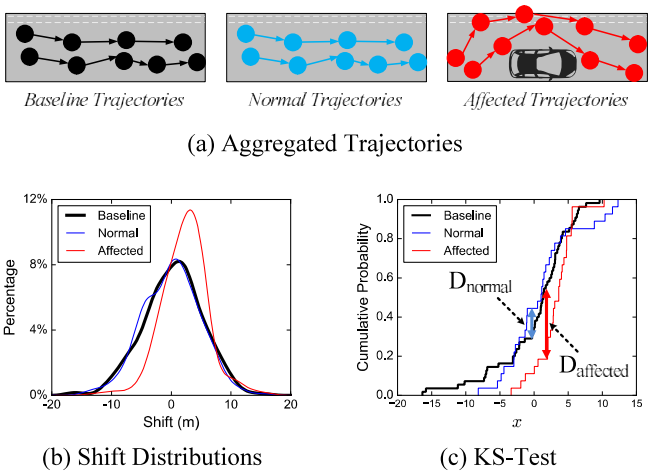
4 ILLEGAL PARKING DETECTION

4.1 Overview

Challenges. With the massive, high quality, and pre-processed bike trajectories, detecting illegal parking events on a road segment is still a very hard problem:

(1) *No labelled data.* We do not have large scale labels for illegal parking events, which makes it hard to apply the conventional classification models directly.
(2) *Complex illegal parking events.* The scenarios of illegal parking events are various, even on the same road, as they appear with different numbers and in different positions.
(3) *Variant individual behaviours.* Users have different riding preferences and behaviours, which makes it unstable to infer illegal parking events with an individual trajectory.
(4) *GPS inaccuracy.* The accuracy of GPS readings is limited, and there can be some minor random shifts.

Intuitions. To overcome these challenges, four corresponsive intuitions are employed: 1) it is hard for us to collect a large scale dataset with illegal parking events, but it is relatively easier to identify negative labels from the dataset (i.e., the normal trajectories) with some heuristics. 2) comparing to the complicated trajectory characteristics with illegal parking events, the trajectory characteristics without illegal parking events are more stable and easier to indentify; 3) Instead of testing a trajectory individually to see if it is impacted by an illegal parking event, we aggregate all of the trajectories in a time period (i.e., one hour), and extract the overall features; and 4) as the GPS accuracy is highly related with the



(a) Aggregated Trajectories

(b) Shift Distributions (c) KS-Test

Figure 10: Main Ideas of Illegal Parking Detection.

build-up proximity [22], we build the baseline models to describe the normal trajectory features on each individual road segment.

Main Ideas. With the above intuitions, the shift distribution of the aggregated trajectories is used as the feature to evaluate the existence of illegal parking events on a road segment, as it is a direct result and significantly obvious from our intuition validation experiments in Figure 3.

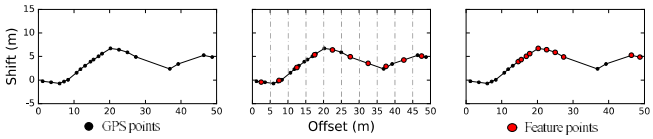
To calculate the difference of shift distributions between the aggregated normal trajectories (or the baseline model) and the evaluation trajectories at the same road segment (i.e., demonstrated in Figure 10a and 10b), the Kolmogorov-Smirnov test (or KS test) is used [8]. Figure 10c illustrates the semantic meaning of KS test statistics, where the shift distributions are more similar if there is no illegal parking events on the evaluation trajectories. Then, we set one threshold to determine if the two sets of trajectories are from the same distribution (i.e., evaluation trajectories are in the same scenario as the normal condition) to infer if there are illegal parking events. One threshold is used here, as the impact of trajectory shift from illegal parking events is the same across the whole city (i.e., around the width of a vehicle). Finally, we evaluate the test results of different threshold values to determine the most effective threshold based on the labelled illegal parking events that we collected.

The following sections describe the details on: 1) building a baseline trajectory distribution model for each road segment; 2) extracting features from evaluation trajectories; and 3) performing the distribution test-based evaluation and selecting the threshold to make the detection.

4.2 Baseline Trajectory Modelling

A baseline trajectory model is built at each road segment to capture the shift distribution for trajectories at the normal scenario (i.e., without illegal parking events). Two heuristics are used:

Naive Baseline Model. A naive baseline uses the shape of a road directly. The assumption of this approach is that if the trajectories travel without the impact of illegal parking events, they should travel perfectly along the shape of the road segment. We use zero



(a) Original Trajectory (b) Average Shift Extraction (c) Top Shift Extraction

Figure 11: Examples of Trajectory Feature Extraction

mean Gaussian distribution to simulate the trajectory shifts in normal scenarios.

Night Time Baseline Model. This baseline assumes the bike trajectories at night (e.g., 11:00 PM to 7:00 AM, in our implementation), in the most cases, travel without the impact from illegal parking events. To overcome the challenges that 1) it is possible to have occasional overnight illegal parking events on the street, and 2) there are very limited number of trajectories travelling during that time period (less than 5% in the dataset), we aggregated shifts of trajectories on each road segment for a very long time period (i.e., over six months), when constructing baseline models, to minimize the impacts from the above challenges.

We noted that it is still possible to have "regular" illegal parking events during the night time at some road segments, e.g., near a dense residential area. However, we consider them as common knowledge that can be discovered easily, and are not the focus in our technique.

4.3 Evaluation Trajectory Feature Extraction

This step extracts the features from a set of evaluation trajectories, aggregately, as the individual trajectory is relatively unstable. To ensure a fair sampling from the each trajectory in the trajectory set, two tasks are performs: 1) the trajectories on the road are further segmented (as 50 meters) based on their GPS offsets, to minimize the case that the shift distribution incurred by an illegal parking event is neutralized by the no-illegal parking portion in a very long road segment; 2) GPS shifts are re-sampled uniformly (e.g., one GPS point every 5 meters) to avoid the case that the shift distribution is dominated by a few highly sampled or very slow trajectories, as there are significant difference between users' behaviors and devices. Then, two methods are proposed to extract the features from evaluation trajectories.

Average Shift Extraction. This method calculates one average shift value based on the all the GPS points within the 5-meters' range of their offsets. As in Figure 11b, the black dots are the original GPS points, and the red dots in each offset range (marked by the dotted vertical lines) are the calculated average shift values. Finally, all calculated shifts are returned as the features.

Top Shift Extraction. This method extracts the top shift values for each trajectory, where the top-10 samples are extracted for each 50-meter road segment in our implementation, as illustrated in Figure 10c. The intuition here is to avoid missing any large shifts caused by a potential illegal parking event. As demonstrated in Figure 11, *average shift extraction* doesn't include the highest shift values, which potentially are the results of illegal parking events.

4.4 Distribution Test-based Detection

We use Kolmogorov-Smirnov test (or KS test) statistic on the shift samples from evaluation trajectories and the baseline trajectories to determine if the two samples are drawn from the same distribution. The intuition is that, if the two shift samples are similar enough, then they belong to the same scenario (i.e., without illegal parking events on the road). Otherwise, we consider them as affected by the illegal parking events.

KS-Test Statistic Calculation. KS statistic essentially calculates the maximum deviation between two empirical cumulative distribution functions, as demonstrated as the D_{normal} and D_{affected} in Figure 10c:

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)| \tag{1}$$

where $D_{n,m}$ is the KS statistic, $F_{1,n}$ and $F_{2,m}$ are the empirical cumulative shift distributions of the baseline model and the features of the evaluation trajectories, m and n are the number of shift samples, and *sup* is the supremum function.

Threshold Selection. We reject the assumption that the two samples are from the same distribution (i.e., essentially no illegal parking event) based on the following equation:

$$D_{n,m} > c(\alpha)\sqrt{\frac{n+m}{nm}} \;\&\; c(\alpha) = \sqrt{-\frac{1}{2}ln(\frac{\alpha}{2})} \tag{2}$$

where α is the probability reflecting if two samples are from the same distribution in KS test. α also is used as the threshold used to reject the assumption that the two samples are from the same distribution (or essentially deciding if the road is with illegal parking events). Instead of using the standard probability threshold, e.g., $\alpha = 0.05$, we test a series of different α values based on an illegal parking data set that we collected to select the most effective one. The details of the threshold selection are in Section 6.2.

5 SYSTEM DEPLOYMENT

In this section, we first describe the system interface. After that, we present the detailed deployment on the cloud.

5.1 System Interface

Figure 12 shows the interface of our system [1], with three views:

Parameter View. The user first selects an area for detection (in this demo, we have Chaoyang District and Haidian District). After that, the user selects a date and time periods for illegal parking events detection.

Result View. This view shows a list of road segments with possible illegal parking events, ordered by their KS statistics.

Map View. Users can see the road segments in the *result view* with different colors indicating the severity of illegal parking events (e.g., the black and purple colors mean severe, and the green color is light). The marker on the map shows the selected road segment.

5.2 Cloud Deployment

To improve the response time for detecting the illegal parking events based on massive trajectories over the whole city, the system is deployed based on a parallel computing platform, i.e.,

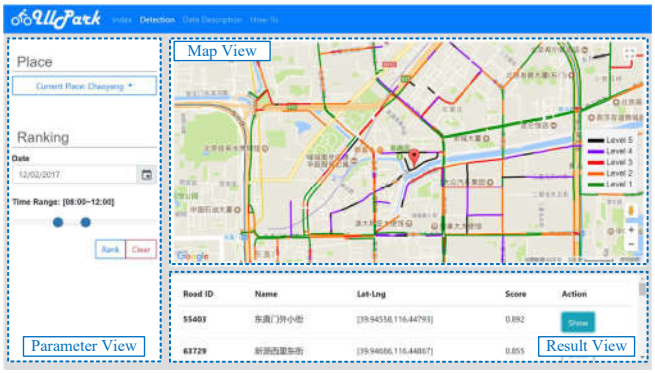


Figure 12: System Interface.

Apache Storm. Figure 13 gives an overview of our system deployment on the cloud, with two phases:

Warm Up. In this phase, the *command spout* sends out a set of road segment IDs to the worker nodes using `ShuffleGrouping` method (i.e., random distribution). For each worker node, they will load the baseline models of the assigned road segments from our trajectory data storage (i.e., MongoDB cluster). After all the baseline models are loaded, the worker node notifies the report bolt. When all the worker bolts are ready, the *warm up* phase ends. In this work, the bike trajectories in MongoDB are map-matched offline based on a trajectory preprocessing framework [26] for mining the historical illegal parking hotspots. Later, by adopting the streaming based map-mathcing system, e.g., [4], the system can be easily extended to support real-time map-matching and detection in a city.

Servicing. In this phase, a user can input a request to the *command spout* to evaluate a set of trajectories in a given temporal range (e.g., the last hour). The temporal range is sent to the *worker bolts* with `AllGrouping` method (i.e., broadcast). For each *worker bolt*, it queries the trajectory storage (i.e., MongoDB cluster) to retrieve the trajectories passed during the given temporal range of its preloaded road segment. Then, it performs a KS-test to determine if the road segment has illegal parking event. Finally, the detection results of all the road segments are sent to *report bolt* to provide an overall ranking of illegal parking impact at the given time period.

6 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the effectiveness and efficiency of our system. We first describe the real dataset used in the paper. Then, we give experiment results to select the proper threshold α used in the KS-test. Then, effectiveness comparisons between different baseline solutions are provided. After that, we test the efficiency performance of our system based on different sizes of Storm clusters. Finally, a set of real case studies are presented to demonstrate the effectiveness of our solution.

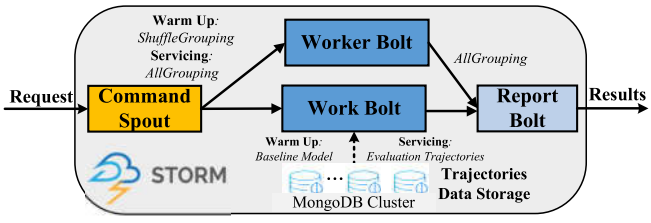


Figure 13: Cloud-based Deployment.

6.1 Datasets

Road Networks. Road network of Beijing, China is from Open Street Map [2] with 377,559 nodes and 501,462 edges.

Mobike Trajectories. Each Mobike trajectory contains a bike ID, a user ID, a temporal range of the trajectory, a pair of start/end locations, and a sequence of intermediate GPS points.

The dataset used in the paper is the full Mobike trajectory data in the City of Beijing, with the time span of 08/01/2017 - 2/08/2018, the spatial distribution is shown in Figure 2a [3].

Ground Truth Labels. We collected a set of ground truth labels for illegal parking events in both Chaoyang and Haidian District, in Beijing. Each record contains a road ID, a timestamp, a photo, and a label to indicate the presence of illegal parking events.

The dataset covers 32 roads, and spans over 18 days (12/26/2017 - 12/30/2017 in Haidian, and 01/12/2018 - 02/09/2018 in Chaoyang). Overall, the 454 ground truth labels are collected, with 159 records labelled as positive (i.e., with illegal parking events).

6.2 Effectiveness Evaluation

In this subsection, we evaluate the effectiveness of the proposed detection model. We first introduce the process to select the most effective threshold for KS-test. Then, we compare our algorithm with three baseline methods. Finally, we study the effectiveness of detection model with different numbers of trajectories.

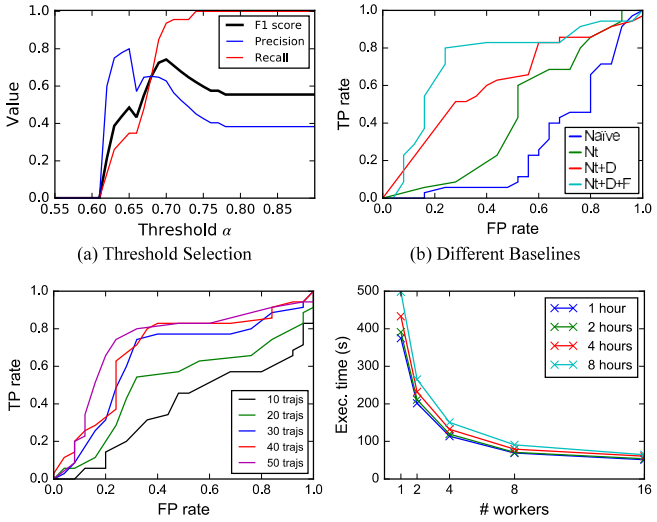
Threshold Selection. In our implementation, we tried all possible threshold probability α in our KS-test from 0 to 1 with the step size of 0.01. To determine the most effective threshold, we test the detection result with the ground truth labels. A F1 score is calculated to reflect the effectiveness of different threshold values. In detail, we count the numbers of True Positives N_{TP} (i.e., correct identification of positive labels), False Positives N_{FP} (i.e., incorrect identification of positive labels), and False Negatives N_{FN} (i.e., incorrect identification of negative labels) to calculate F1-score:

$$\begin{aligned} P &= N_{TP}/(N_{TP}+N_{FP}), \\ R &= N_{TP}/(N_{TP}+N_{FN}), \\ F_1 &= 2PR/(P+R). \end{aligned} \tag{3}$$

Figure 14a presents the F1 scores with the corresponding *precision* P and *recall* R with respect of different threshold values α: 1) when α is close to zero, all the test data are labelled as negative, so both precision and recall are 0; 2) with the increase of α, more and more instances are labeled as positive, and we identify that $\alpha = 0.71$ is the best selection as F1-score reaches the maximum of 0.73. As a result, we use $\alpha = 0.71$ as the threshold in our system.

[2] https://www.openstreetmap.org/

[3] The detailed statistics of the trajectory dataset are not disclosed in the paper, due to the request from Mobike.



(a) Threshold Selection (b) Different Baselines

(c) Impact of Different # Trajectories (d) Efficiency Experiments

Figure 14: Effectiveness & Efficiency Experiments.

Compare with Baselines. To achieve a comprehensive comparison with different baseline approaches, we plot the Receiver operating characteristic (ROC curves) for each method. ROC curve illustrates the diagnostic ability of a binary classifier, which plots the true positive rate $TPrate = N_{TP}/(N_{TP} + N_{FN})$ and the false positive rate $FPrate = N_{FP}/(N_{FP} + N_{TN})$ of a model with different parameter settings (i.e., threshold α). A method is better, if its AUC (Area Under Curve) is larger.

In this experiment, we compare our deployed solution (i.e., *Nt+Dir+T*) with three baseline methods:

• **Naive.** It uses the shape of the road segment directly as baseline model, i.e., the shift distribution is set as a Gaussian Distribution with zero mean and 2σ value is set as 10 meters. *Average shift extraction* is used for processing the evaluation trajectories.

• **Nt.** It uses the aggregated night time trajectories to build the baseline model. The features are extracted based on *average shift extraction* method.

• **Nt+Dir.** In addition to use the night time trajectories as the baseline mode, it also filters the reversed trajectories in each road segment. The trajectory features are extracted based on *average shift extraction* method.

• **Nt+Dir+T.** It uses the same baseline model as *Nt.+Dir*, while *top shift extraction* method is employed to extract the features of evaluation trajectories.

Figure 14b presents the ROC curves of the 4 baseline methods by varying the threshold α of KS-test. In the measurement of AUC, our deployed method, i.e., *Nt+Dir+T* outperforms other three methods significantly, since it considers both directional information and top shift features. The method considering the directional information and removing all the reverse trajectories is much better than the others, which demonstrates the necessity of reverse trajectory removal in the *pre-processing* component. Also in the *Naive*, the TP rate is always smaller than FP rate, which implies the inaccuracy of GPS position cannot be neglected.

Different Trajectory Numbers. We study the influence of the detection effectiveness with different numbers of trajectories in the evaluation trajectory feature extraction step. In this experiment, we only choose a portion of labelled road segments with the number of trajectories over 50 in one hour time range, and down samples the dataset randomly to mimic the road with different numbers of trajectories from 10 to 50.

Figure 14c illustrates the ROC curves of our deployed method with different numbers of trajectories used for detection. From the figure we can see that the detection performance increases, when more trajectories can be used. Moreover, when the number of trajectories can be used for detection is less than 20, the detection accuracy is unstable. It confirms the necessity of using the crowd-wisdom to overcome the impacts from skewed individual riding behaviors and the limited GPS accuracy. Finally, the performance of detection model increases slightly when the number of trajectories is over 30. It shows that as long as it has enough number of trajectories (e.g., 30) in an hour on a road segment, our method can provide a relatively accurate detection result.

6.3 Efficiency Evaluation

The system response time is also tested to show the efficiency of our cloud-based deployment. The experiments are performed in Haidian District, which contains 10% of the total trajectories in Beijing. We tested the different numbers of requests (i.e., to evaluate 1, 2, 4, and 8 hours of trajectories in the area, the evaluation requests with more than one hour perform the one-hour detection multiple times) with different numbers of worker nodes in Storm.

Figure 14d gives the results of system response time with different settings. We have the following two observations: 1) the execution time decreases significantly by nearly 50% when adding the number of workers from 1 to 2 and 2 to 4, since the detection tasks are distributed among the worker nodes; and 2) when the number of workers increases further to 16, the performance gain is much less. This is due to the communication overhead of worker nodes. In terms of different sizes of trajectory data, the response time difference is relatively minor, as the MongoDB cluster provides an efficient trajectory data retrieval interface.

6.4 Case Studies

We conduct real world case studies to validate the effectiveness of our detection model. Moreover, we get some interesting observations by exploring the temporal variance of KS statistics.

Overall Rankings. To reflect the severity of the illegal parking events on the roads, we rank them based on the daily average hours with illegal parking events. To validate the correctness of our result, we conduct an on-site case study in the area of Figure 15). The area marked with the red dotted lines suffers from sever illegal parking events, based on our calculations. The reason becomes clear, when we got there. It is a very crowded area, with a lot of foreign embassies and high-end restaurants, but with very limited parking spaces. As a result, many people have to leave their cars in the bike lanes or on the pedestrian crossing (as demonstrated in the left portion of the figure). On the other side, at the east side of the East 3rd ring road, the overall rankings of the green road segments are very low. It is because the POI distribution there is very sparse, with only two large hotels, and the Agriculture Exhibition Center. All of these places are facilitated with very large parking

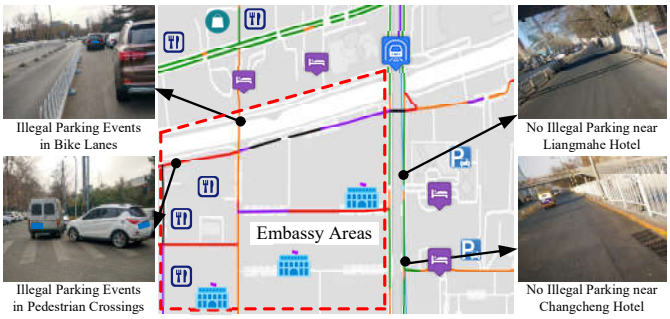


Figure 15: Case Study of Overall Ranking.

lots. As a result, as shown in the right portion of the figure, the road are clear without any illegal parking events.

Impact of Rush Hours. The KS statistic varies differently at different times on a road segment, which indicates the the temporal differences of illegal parking events. We studied a road segment near Liangmaqiao Subway station, whose KS statistic is significantly higher during the rush hour, as shown in Figure 16a. We visit the area multiple times and notice that, during the evening rush hours, we notice much more illegal parking events: either from some uber/didi drivers waiting for their customers from the subway station, or some people who visit the nearby restaurants and didn't find a parking space.

Impact of Holiday Events. It is also interesting to observe that a local park (Wanghe Park) near Wangjing area, is significantly impacted by different days. We notice that the road in Figure 16b has significantly different KS statistics at the same time between weekdays and weekends. After we got there, we notice that the park is holding a snow festival. A lot of children practice skiing there, as the PyeongChang Winter Olympic Games is approaching. As a result, there are much more illegal curbside vehicle parking events during the weekends, as some of the parents can not find a parking space nearby.

7 RELATED WORK

We summarize the related works in three main areas: 1) urban computing, 2) trajectory data mining, and 3) urban crowd sourcing.

Urban Computing. Urban computing [39] aims to address different problems in the city. For example, [7, 28, 35] predict the taxi demand to enable smart scheduling, and reduce energy wasting. [5, 31, 36, 40] try to understand human mobility patterns from check-ins. Illegal parking detection is also an important issue in the urban computing, where the most of the existing methods are based on video surveillance [12, 14], with limited coverage in a city. However, the system we proposed utilizes an ubiquitous approach.

Trajectory Data and Mining. Our technique is highly related with the trajectory data mining. Based on the massive bike trips, [3] provides bike lane planning recommendations to the goverment. And the frequent route mining [6, 17, 20, 23] is beneficial to city planners and provides a guidance for congestion analysis. In addition, [18] finds top-k influential locations that cover as much trajectories. In addition, to improve person's travel experience, trajectory based solutions for travel time estimation [19, 29]

Figure 16: Observations with Temporal Differences.

and reachability query [30, 32] are proposed. The closest projects to us are the trajectory anomaly detection, which aims to find trajectory that is dissimilar to the majority of the others. [16, 34] are based on classification model, while others [10, 15] compare trajectory similarity with history. However, the existing trajectory anomaly detection methods only focus on the high-level route-based difference, while in our problem settings, the difference between trajectories is more subtle at the same road segment.

Urban Crowd Sourcing. Essentially, we take the advantage of the massive Mobike users in a city to perform the detection task, which makes us very related to the crowd sourcing techniques. For example, [33] quantifies the fragility of cities through detecting the delay in commuting activities using GPS data collected from smartphones. [24, 25] infer noise levels for locations by smartphone users. [9, 11] identify potholes or classify road quality from vehicle's accelerometer data. Different from the above works, we focus on the problem of illegal parking detection.

8 FUTURE WORK

In this paper, we focus on the illegal parkings happening at curbsides, and propose a data-driven illegal parking detection framework based on the data mining results from massive sharing bike trajectories. However, just using the bike trajectories is not the silver bullet to solve the illegal parking detection. In the future, we plan to extend the work to address the following two problems: 1) distinguishing illegal parkings from other events that influence the bike trajectories. Figure 17a gives an example of road maintaining, where the riders are forced to vehicle lanes, which leads

(a) Road Maintenance (b) Parking on Vehicle Lanes

Figure 17: Future Work.

the changes in trajectory patterns. Also, some of the curbsides can provide legal parking space; and 2) detecting the illegal parkings out of curbsides. As is shown in Figure 17b, the parked cars blocks the partial of the vehicle lanes. However, they do not affect to the passing-by bike trajectories, which may lead to false negative result in our settings.

9 CONCLUSION

We propose a novel and ubiquitous approach to detect illegal parking events with massive and fine-grained sharing bikes' trajectories. Based on the unique properties of the bike trajectories, we design a comprehensive *pre-processing* component to overcome numerous challenges in data cleaning, map-matching and indexing. In the *illegal parking detection* component, we employ a distribution test-based method to determine if the set of evaluation trajectories have the same aggregated shift distribution with the baseline models (i.e., without illegal parking events). The system is deployed on the cloud and used internally. Extensive experiments are performed on a large scale Mobike data. Based on over 400 ground truth labels, our model achieves an F1 score of 0.73, which outperforms all the other baseline approaches. Finally, real world case studies are conducted, which provides us with many insights.

ACKNOWLEDGEMENT

We thank Beijing Mobike Technology Co., Ltd. for providing the Mobike trajectories.

This work was supported by the National Natural Science Foundation of China Grant No. 61672399 and No. U1609217.

Yanhua Li was supported in part by NSF CRII grant NSF CNS-1657350 and a research grant from DiDi Chuxing Inc.

REFERENCES

[1] 2017. IllParking System. http://illparking.urban-computing.com/.
[2] 2017. Mobike Awarded with UNâĂŹs Top Environmental Prize. https://pandaily.com/mobike-awarded-uns-top-environmental-prize/.
[3] Jie Bao, Tianfu He, Sijie Ruan, Yanhua Li, and Yu Zheng. 2017. Planning Bike Lanes Based on Sharing-Bikes' Trajectories. In *SIGKDD*. ACM.
[4] Jie Bao, Ruiyuan Li, Xiuwen Yi, and Yu Zheng. 2016. Managing massive trajectories on the cloud. In *SIGSPATIAL GIS*. ACM, 41.
[5] Jie Bao, Defu Lian, Fuzheng Zhang, and Nicholas Jing Yuan. 2016. Geo-social media data analytic for user modeling and location-based services. *SIGSPATIAL Special* 7, 3 (2016), 11–18.
[6] Zaiben Chen, Heng Tao Shen, and Xiaofang Zhou. 2011. Discovering popular routes from trajectories. In *ICDE*. IEEE, 900–911.
[7] Zheng Dong, Cong Liu, Yanhua Li, Jie Bao, Yu Gu, and Tian He. 2017. REC: Predictable Charging Scheduling for Electric Taxi Fleets. In *RTSS*. IEEE.
[8] Zvi Drezner, Ofir Turel, and Dawit Zerom. 2010. A modified Kolmogorov–Smirnov test for normality. *Communications in Statistics Simulation and Computation* (2010), 693–704.
[9] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. 2008. The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. In *MobiSys*. ACM, New York, NY, USA, 29–39.
[10] Yong Ge, Hui Xiong, Chuanren Liu, and Zhi-Hua Zhou. 2011. A taxi driving fraud detection system. In *ICDM*. IEEE, 181–190.
[11] Marius Hoffmann, Michael Mock, and Michael May. 2013. Road-quality classification and bump detection with bicycle-mounted smartphones. In *UDM*. CEUR-WS. org, 39–43.
[12] MX Jiang, HY Wang, and FS Mu. 2012. Illegal parking detection algorithm based on video surveillance. *Computer Engineering* 38, 19 (2012), 151–153.
[13] Markos Kladeftiras and Constantinos Antoniou. 2013. Simulation-based assessment of double-parking impacts on traffic and environmental conditions. *Transportation Research Record: Journal of the Transportation Research Board* 2390 (2013), 121–130.
[14] Jong Taek Lee, Michael Sahngwon Ryoo, Matthew Riley, and JK Aggarwal. 2009. Real-time illegal parking detection in outdoor environments using 1-D transformation. *TCSVT* (2009), 1014–1024.
[15] Po-Ruey Lei. 2016. A framework for anomaly detection in maritime trajectory behavior. *KAIS* 47, 1 (2016), 189–214.
[16] Xiaolei Li, Jiawei Han, Sangkyum Kim, and Hector Gonzalez. 2007. Roam: Rule- and motif-based anomaly detection in massive moving object data sets. In *SDM*. SIAM, 273–284.
[17] Xiaolei Li, Jiawei Han, Jae-Gil Lee, and Hector Gonzalez. 2007. Traffic density-based discovery of hot routes in road networks. In *SSTD*. Springer, 441–459.
[18] Yuhong Li, Jie Bao, Yanhua Li, Yingcai Wu, Zhiguo Gong, and Yu Zheng. 2016. Mining the most influential k-location set from massive trajectories. In *SIGSPATIAL GIS*. ACM, 51.
[19] Yang Li, Dimitrios Gunopulos, Cewu Lu, and Leonidas Guibas. 2017. Urban Travel Time Prediction using a Small Number of GPS Floating Cars. In *SIGSPATIAL GIS*. ACM, 3.
[20] Wuman Luo, Haoyu Tan, Lei Chen, and Lionel M Ni. 2013. Finding time period-based most frequent path in big trajectory data. In *SIGMOD*. ACM, 713–724.
[21] Khaled Mahmud, Khonika Gope, and Syed Mustafizur Rahman Chowdhury. 2012. Possible causes & solutions of traffic jam and their impact on the economy of Dhaka City. *JMS* 2, 2 (2012), 112.
[22] Marko Modsching, Ronny Kramer, and Klaus ten Hagen. 2006. Field trial on GPS Accuracy in a medium size city: The influence of built-up. In *WPNC*, Vol. 2006. 209–218.
[23] Dev Oliver, Shashi Shekhar, Xun Zhou, Emre Eftelioglu, Michael R Evans, Qiaodi Zhuang, James M Kang, Renee Laubscher, and Christopher Farah. 2014. Significant route discovery: A summary of results. In *ICGIS*. Springer, 284–300.
[24] Zhaokun Qin and Yanmin Zhu. 2016. NoiseSense: A Crowd Sensing System for Urban Noise Mapping Service. In *ICPADS*. IEEE, 80–87.
[25] Rajib Kumar Rana, Chun Tung Chou, Salil S. Kanhere, Nirupama Bulusu, and Wen Hu. 2010. Ear-phone: An End-to-end Participatory Urban Noise Mapping System. In *IPSN*. ACM, 105–116.
[26] Sijie Ruan, Ruiyuan Li, Jie Bao, Tianfu He, and Yu Zheng. 2018. CloudTP: A Cloud-based Flexible Trajectory Preprocessing Framework. In *ICDE*. IEEE.
[27] YingLi Tian, Rogerio Schmidt Feris, Haowei Liu, Arun Hampapur, and Ming-Ting Sun. 2011. Robust detection of abandoned and removed objects in complex surveillance videos. *IEEE Transactions on Systems, Man, and Cybernetics* 41, 5 (2011), 565–576.
[28] Dong Wang, Wei Cao, Jian Li, and Jieping Ye. 2017. DeepSD: supply-demand prediction for online car-hailing services using deep neural networks. In *ICDE*. IEEE, 243–254.
[29] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel Time Estimation of a Path using Sparse Trajectories. In *KDD*. ACM.
[30] Di Weng, Heming Zhu, Jie Bao, Yu Zheng, and Yingcai Wu. 2018. HomeFinder revisited: finding ideal homes with reachability-centric multi-criteria decision making. In *CHI*. ACM.
[31] Fei Wu and Zhenhui Li. 2016. Where did you go: Personalized annotation of mobility records. In *CIKM*. ACM, 589–598.
[32] Guojun Wu, Yichen Ding, Yanhua Li, Jie Bao, Yu Zheng, and Jun Luo. 2017. Mining Spatio-Temporal Reachable Regions over Massive Trajectory Data. In *ICDE*. 1–12.
[33] Takahiro Yabe, Kota Tsubouchi, and Yoshihide Sekimoto. 2017. CityFlowFragility: Measuring the Fragility of People Flow in Cities to Disasters using GPS Data Collected from Smartphones. *IMWUT* 1, 3 (2017), 117.
[34] Wanqi Yang, Yang Gao, and Longbing Cao. 2013. TRASMIL: A local anomaly detection framework based on trajectory segmentation and multi-instance learning. *Computer Vision and Image Understanding* 117, 10 (2013), 1273–1286.
[35] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. In *AAAI*.
[36] Hongzhi Yin, Zhiting Hu, Xiaofang Zhou, Hao Wang, Kai Zheng, Quoc Viet Hung Nguyen, and Shazia Sadiq. 2016. Discovering interpretable geo-social communities for user behavior prediction. In *ICDE*. IEEE, 942–953.
[37] Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, and Guang-Zhong Sun. 2010. An interactive-voting based map matching algorithm. In *MDM*. IEEE Computer Society, 43–52.
[38] Yu Zheng. 2015. Trajectory data mining: an overview. *TIST* 6, 3 (2015), 29.
[39] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *TIST* 5, 3 (2014), 38.
[40] Yuan Zhong, Nicholas Jing Yuan, Wen Zhong, Fuzheng Zhang, and Xing Xie. 2015. You are where you go: Inferring demographic attributes from location check-ins. In *WSDM*. ACM, 295–304.