Exploiting Mobile Sensing to Enable Automated Crowdsourced Parking Availability Sharing

Tingxin Yan[§], Baik Hoh[†], Deepak Ganesan[§]
[§]Dept. of Computer Science, University of Massachusetts, Amherst MA 01003
[†]Nokia Research Center, Palo Alto CA 94304
{yan, dganesan}@cs.umass.edu[§], {baik.hoh}@nokia.com[†]

ABSTRACT

Limited parking resource in urban areas causes severe problems including traffic congestion, environmental pollution, driver anxiety, and many others. These concerns would be significantly alleviated had real time parking availability information were available to drivers. Compared with expensive infrastructure-based approaches, crowdsourcing-based parking availability systems, which exploits smartphone users to report available parking spots, has begun to attract attention from both academia and industry due to its large scale, agility, and low cost. However, existing crowdsourced parking information systems rely on manual reports from participants, which suffers from cumbersome manual operations as well as inaccurate reports due to human errors.

In this paper, we present SENPARK, a mobile platform that combines crowdsourcing and mobile sensing together to enable accurate and automated available parking information reporting from mobile users. Compared with existing crowdsourced parking availability platforms, SENPARK has two significant contributions. First, it presents a parking detection algorithm that enables users to check in parking locations automatically. Second, it presents a leaving time estimation algorithm that enables users to automatically report their leaving events a few minutes ahead.

With real-world experiments consisting of 5 participants over 2 months, we show that SENPARK can 1) report parking events of users with an accuracy of over 90%, 2) forecast leaving events a couple minutes ahead with an accuracy of over 95%, and 3) recognizing over 95% malicious users.

1. INTRODUCTION

Parking in crowded urban areas is a precious resource and drivers spend substantial amounts of time locating empty parking spots. Metropolitan cities, such as San Francisco and New York City in particular, have a pressing problem due to limited parking in downtown areas, both for street as well as garage parking [14]. Making matters worse, information about parking availability is typically unavailable to drivers. Studies by the US Department of Transportation have reported that parking patrons "often do not know where the best parking locations are", and "most importantly, whether a parking place will be available when they arrive" [12].

The chasm between demands versus supply of parking spots causes a spectrum of environmental, health and safety issues. Drivers who keep vehicles on the road circling for parking could lead to lengthy queues of vehicles causing severe traffic congestions. In addition, the frequent switches between acceleration and braking while circling not only generates significant amount of automobile emissions [1], but increases the stress level of drivers and has been reported to increase road rage and accidents [4].

These concerns have led to significant efforts to design online real-time parking information systems that can provide accurate and real time information about parking availability. For example, SFPark is a pilot project to monitor real-time parking availability in San Francisco by deploying a massive network of sensors [13]. While such parking information systems can help direct drivers to available parking locations, they are in their pilot stages and face daunting scaling and budgetary challenges given the vast volume of street parking in U.S. cities. Continuous monitoring of street parking requires installation of occupancy sensors on hundreds of thousands of parking spots or parking meters, and a vast wireless infrastructure to obtain and transmit sensing data in a reliable manner.

The difficulties in deploying continuous-sensing based parking infrastructure has led to increased interest in the use of crowdsourcing using mobile phones. Several mobile applications such as Google's OpenSpot [8] and PrimoSpot [2] were recently released, with the intent of using general public to locate empty parking spots. Compared with infrastructure-based approaches such as SFPark [13], crowdsourcing-based approaches offer higher agility, lower cost, and larger coverage. Recent research has also shown that using distance sensors attached to taxis [11] for obtaining parking availability information is promising, although it is a more expensive and less scalable approach than using mobile phones.

A fundamental limitation in all these systems is that they rely on mobile users to manually input parking availability information. Dependency on human operations would not only degrade user participation, or increase the incentives needed, due to the cost of human operations, but incur inaccurate or error reports. For example, mobile users are easily forget about pressing certain buttons to report parking availability information, or had some incorrect operations.

In this paper, we address this limitation of crowdsourced parking systems by designing SENPARK, a parking availability sharing system for mobile users to report available-now and available-soon parking spots with no human operations. Mobile users can leave SENPARK running in the background, and the SENPARK service can automatically detect parking events, predict the leaving events of mobile users and broadcast available-soon information, and detect pulling out events and confirm available-now events.

Although the vision of automatic parking availability report is intriguing, a practical realization of such a system presents several hurdles. First, such a system should accurately detect motion states of mobile users, such as walking and driving, and reliably infer parking or leaving events triggered by the change of motion states. Second, such a system should be able to predict that mobile users are coming back to parked spot to forecast their leaving events accurately. Third, accurate localization should be also supported since the system needs to provide the location of available parking spots without manual verification steps as well. Fourth, such a system should be robust against malicious users whose goal is solely to maximize their gain by masquerading as fake participants.

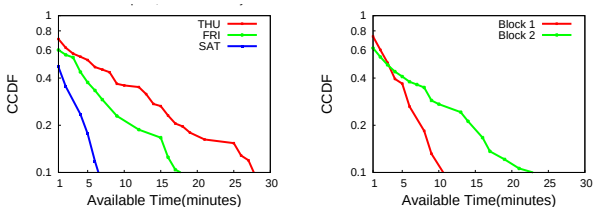
SENPARK addresses these challenges by using a novel combination of sensing and inference methods. We developed an activity recognition algorithm that uses only accelerometer sensor to accurately detect the motion state of mobile users. We also developed a geofence-based location trace prediction algorithm that can forecast leaving events of mobile users. We also proposed an activity recognition based scheme to detect malicious users in the system thus enhance the robustness of the system. We show that:

- Our geofencing-based leaving events forecasting algorithm can achieve 95% precision and recall.
- Our accelerometer-based parking events detection algorithm can achieve 95% accuracy with 30 second delay.
- Our sensing-based approach can detect malicious users with close to 100% accuracy when they are pedestrians and over 95% when they are motorists.

The rest of this paper is organized as follows. Section 2 introduces the overview of SENPARK and share our motivation and challenges. Then we explain three core building blocks to tackle these challenges in Sections 3, 4, and 5. In Section 6, we evaluate the performance of SENPARK system and discuss its limitations. Finally we summarize related work in Section 7 and conclude the paper in Section 8.

2. SENPARK OVERVIEW

The goal of SENPARK is to provide a crowdsourcing parking availability tool which allow mobile users to provide accurate parking availability information with minimized user operations.



(a) Temporal Behavior (b) Spatial Behavior

Figure 1: Spatial and Temporal Dependency of Parking Availability in San Francisco

The first question SENPARK addresses is the usefulness of parking availability information. Existing crowdsourced systems, such as Google's OpenSpot [8] and PrimoSpot [2], allow mobile users to provide availability of parking spots that are already empty. The key disadvantage of such availability information is that the life time of an empty parking spot is usually very short in urban areas. We have conducted a field study in San Francisco downtown area, where street-parking supply is far less than parking demand. We monitored the street parking spots in four blocks and recorded the time when each parking spot is taken and released. Our observation is shown in Figure 1(b) and Figure 1(a). The availability of parking exhibits significant temporal and special dependency, and generally speaking, the life time of an empty parking spot in SF downtown can be less than five minutes in busy hours, which makes the parking availability information provided by mobile users very easy to be obsolete. This observation motivates us to seek parking availability information that could last longer, and we argue that instead of providing "available now" information, we could encourage mobile users to provide "available soon" (or "leaving soon") information, which indicates that a parking spot is about to be available within a few minutes. The advantage of "available soon" information is that it gives drivers who receive this information more time to plan their trip, thus the chance of getting a parking spot could be better than only having "available now" information.

In SENPARK, users can be either a contributor who generates parking availability information or a service subscriber who consumes available parking information provided by contributor. Contributors can receive incentives from sharing parking information, but the actual incentive scheme is out of the scope of this paper. The process of how a contributor share the parking availability information is shown in Figure 2. SENPARK only requires a contributor to open the service and let it running at the background. When a contributor parks in a spot, SENPARK automatically recognize the parking event and the location of the spot. When the con-

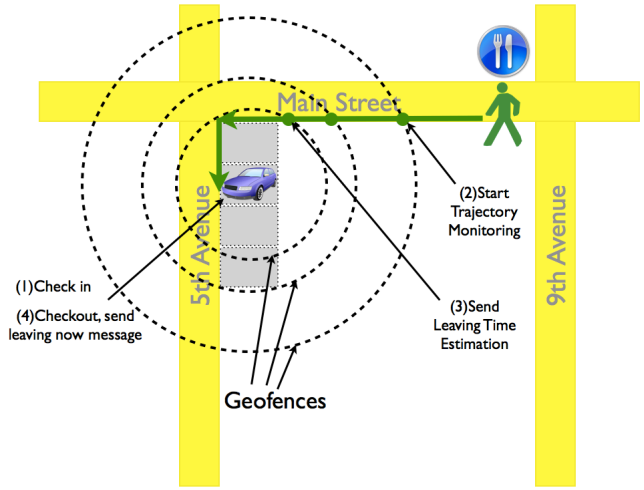


Figure 2: Workflow of SENPARK leaving time estimation scheme

tributor walks back to the parking spot and prepares to leave, SENPARK automatically estimated the leaving time of the contributor and share an "available soon" information to subscribers. When the contributor leaves the parking spot, SENPARK automatically recognize the leaving event and share a "available now" information to subscribers.

Although "available soon" information can greatly prolong the life time of parking availability, how to collect them from mobile user is still challenging. We envision three core challenges in crowdsourcing "available soon" from mobile users. First, how to accurately estimate when a parking spot can be released; second, how to reduce human operation into its minimum during this process; and third, how to design robust systems against malicious users. In Section 3, we describe a geofence based algorithm that can accurately estimate the leaving time of a contributor. In Section 4, we describe an activity recognition based algorithm to accurately detect the parking and leaving events of contributors, and in Section 5, we describe how we use activity recognition to remove malicious users from our system.

3. GEOFENCE-BASED PARKING AVAILABILITY FORECASTING

Available-soon information could greatly enhance the usefulness of parking information, but estimating their own leaving time (i.e., how soon contributors will vacate the spot) is not a trivial task for contributors. Leisure activities such as shopping often have unpredictable schedules because such activities can be easily interrupted by various factors such as chatting with someone or being attracted by merchandises in a shop.

SENPARK proposes an automated leaving time estimation algorithm by analyzing the GPS trajectory of contributors when they are close to their parking spots. The intuition behind our automated leaving time estimation algorithm is that when users walk back to their parking spots and prepare to leave, their trajectories are usually "towards" the parking spot as opposed to random ones that are common for shopping and wandering.

The goal of our algorithm is to predict when contributors will leave (or vacate the parking spot) in advance and forecast the leaving-soon information to surrounding drivers. First, the forecasting of a couple of minutes in advance could increase the value of shared parking information. Note that vacant spots in downtown are so quickly taken (as shown in figure 1(a) and figure 1(b)) that available-soon parking lots can give more time for drivers to prepare themselves than available-now ones. Second, more importantly it gives our service consumers a directed parking guidance such that their hopeless cruising time could be greatly reduced.

Geofences

Before presenting our prediction algorithm, we first need to carefully define what means "when users are close to their parking spots" and what means "users walk towards parking spots". We use geofences to describe the adjacency of contributors when they are close to parking spots. Geofences are basically virtual perimeters for real-world geographic areas. For example, a circle around a shopping mall can be specified as a geofence where only mobile users within the fence should receive certain mobile ads. In SENPARK, we use geofences to trigger our leaving time estimation.

Leaving Time Estimation

The first step of our algorithm is to establish three circled geofences for each contributor, all of which are centered with the geo-coordinates of the parking spot where the contributor is currently parking the vehicle. The first circle has a radius of 150 meters, which corresponds to a distance of two-minute walking distance with an average speed of adult walking. The second geofence has a radius of 225 meters, which corresponds to three-minute walking distance, and 450 meters for the third circle, or 6 minutes walking distance. These three geofences are set up when a contributor checked in, and the precise parking spot is recorded. For the sake of simplicity the geofence radii are fixed in the following sections, but in our implementation we provide users a feature to personalize the geofence with their own paces.

After having three geofences established, SENPARK starts to monitor the GPS location of each contributor. When a contributor walks out of the third geofence, i.e., at least 450 meters away from the parking spot, SENPARK

recognizes that the contributor is far enough from the parking so that the GPS can be duty-cycled to save battery power. In other words, the third circle in SENPARK is an energy fence where GPS sampling rate is much lower outside the fence. In our implementation, we use a duty cycle of 30 second wakeup time and two minute sleep time.

Our leaving time estimation is triggered when our system detects that there is an event of entering to the third circle, i.e., the user has walked back and the distance to the parking spot is less than 450 meters. The GPS sampling rate is increased to be continuous. If the contributor keeps walking towards the parking spot, there will be two more entering events for the second and the first (inner-most) circle, respectively. When these two events occurred, SENPARK system records the intersection point between the contributor's GPS trace and the circle. Let us denote the two intersection points are P_1 and P_2 for first and second circle. Along with the parking spot, denoted as P_0, the three points form a triangle. If the contributor walks towards the parking spot, the angle of $\angle(P_1, P_0, P_2)$ should be small enough, as illustrated in Figure 3(a). On the contrary, if the contributor passes by the parking spot, the angle of $\angle(P_1, P_0, P_2)$ can be large, as illustrated in Figure 3(b). In our algorithm, we set an upper bound for angle $\angle(P_1, P_0, P_2)$ to be $\arccos(150/225)$, which is the corner case where line (P_2, P_1) is in perpendicular to line (P_1, P_0). Only when the angle of $\angle(P_1, P_0, P_2)$ is less then the upper bound, our algorithm thinks that the contributor is walking towards the destination.

Our leaving time estimation finishes when the above trajectory analysis is done. If our algorithm recognizes that the contributor is returning back to the parking spot, SENPARK system will trigger an "available-soon" alert to all system consumers indicating the parking spot will be available in around 2 minutes, which corresponds to the 150 meter radius of the first circle.

4. ACTIVITY RECOGNITION-BASED AUTOMATIC CHECK-IN AND CHECK-OUT

The first task of parking information sharing is to determine the occupation status of a parking spot, including when the spot is taken and released, and what is the location of the spot. In this section, we propose an activity recognition based scheme to automate the process of determine the occupation status of parking spots such that SENPARK contributors can share their parking availability information without any cumbersome manual operations.

SENPARK includes two steps towards automated occupation status recognition — check-in and check-out. The main purpose of check-in step is to detect that a contributor has parked, localize the precise location of the parking spot, and initiate the leaving time estimation algorithm that we described in the previous section. The major goal of check-out step is to detect that a contributor has left the parking spot and announce a "leaving now" message to SENPARK service subscribers that a parking spot is released.

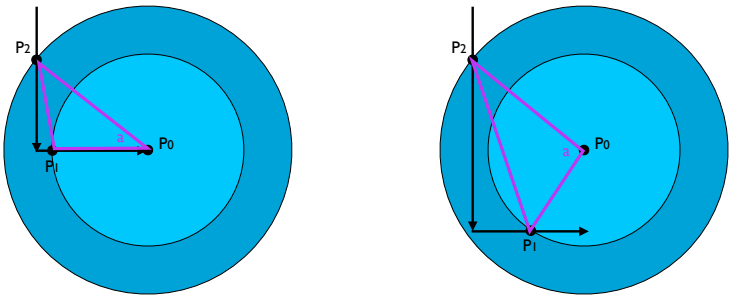
The core of SENPARK automated check-in and check-out are 1) an activity recognition engine that can accurately detect the motion status of mobile users, and 2) a localization engine for accurate parking spot locations.

Recognizing parking event for check-in

Determine that a contributor has parked the car is the foremost task SENPARK needs to complete. SENPARK takes advantages of existing activity recognition engines, such as Jigsaw [10], to accurately detect the motion status of contributors, and use the motion status to infer whether the contributor is in parking state.

Existing activity recognition engines exploits a set of on board sensors, such as accelerometer, compass, and gyroscope sensor, to detect the motion status of humans with high accuracy. For instance, Jigsaw [10] engine claimed that it can distinguish a set of motion status including driving, running, walking, and stationary by primarily using accelerometer sensor. The advantage of Jigsaw is that the energy consumption is very low if only accelerometer is used. In our implementation, we use a similar engine like Jigsaw as a background service to detect the motion status of contributors continuously. The accelerometer sampling process is duty cycled as 25 seconds active and 100 seconds sleep to reduce the energy consumption. When the accelerometer is active, it collects samples with a rate of 20Hz. We first align the accelerometer to remove the gravity so that the accelerometer data can be used to derive the actual speed change of the user. Second, we set up a 25 seconds time window of accelerometer readings to remove jitters. Third, for the 25 seconds accelerometer data, we apply low-pass filter and high-pass filter to get the low- and high-frequency terms, which are useful to distinguish driving activity from others. We then use FFT variance as the major feature for driving state detection. We collect the accelerometer data of the Android smartphone for 7 different cars in driving, walking, and stationary cases to build a training dataset. By using a R45 clustering tree, we were able to classify driving state, walking state, and stationary state with an accuracy of over 90%.

Based on the motion status detected by our activity recognition engine, we use state transition to further infer the parking state. When the activity recognition engine detects a state transition from driving to stationary to walking, it is most likely to be a parking event. SENPARK triggers a check-in message when the state transition is captured, and also initiate localization procedure to capture the accurate location of the



(a) True Example (b) False Example

Figure 3: Geofence examples

parking spot, and leaving time estimation procedure to predict the leaving time of the contributor.

Check-in with accurate parking spot Location

Accurate localization in city areas is a hard problem due to the "urban canyon" effect. Our experiments indicate that GPS localization accuracy in crowded downtown area is highly related to the motion status of users. When users are in moving state, no matter driving or walking, the GPS accuracy is much better than that when users are stationary, either sitting or standing. Furthermore, when users are moving, geometry information, such as road names, can be used to amend GPS readings and improve localization accuracy.

We assume that the GPS is already on when contributors are close to the parking spot. In our current implementation, we specified a set of "hot zones" in San Francisco downtown and use cellular tower positioning to trigger the GPS when contributors enter one of the hot zones. Our envision is that `SENPARK` can be ultimately integrated with smartphone-based navigation software, such as Nokia Navigation, and the GPS can be always on to provide both navigation guidance as well as accurate localization for parking.

Based on our observation, there is a "sweet window" that can achieve the best localization accuracy for parking spot, and the "sweet window" is the last a few samples when a user is about to park, and the first a few samples when a user leaves the car and starts walking. In other words, we need to complete localization as soon as a contributor is checked in. Any time before and after the "sweet window" would leads to large distance between the location of the user and the actual location of the vehicle.

Our localization algorithm takes the last three GPS samples before check-in, and the first three samples right after the walking state is detected, and compute the centroid point from the six GPS locations. The centroid point is used as the location of the parking spot.

Recognizing leaving event for check-out

The last step of `SENPARK` parking status recognition is for contributors to leave the parking spot, and we call it a check-out procedure. The check-out process represents reversed motion status change as check-in process. When users leave the parking spot, the motion status changes from walking to stationary for a short time, and then change to driving state. We still use our activity recognition engine for detecting the state changes, and trigger a "leaving now" message when such changes are detected.

The "leaving now" message is used to confirm the "leaving soon" message that launched by our leaving time estimation algorithm, and also used to indicate that there is actually a parking spot released for `SENPARK` subscribers. Although the life time of "leaving now" message is short, it is still at least not harmful to share such information with other drivers.

Handling recognition delay

Both activity recognition and localization has delays, which may lead to inaccurate recognition or localization results. `SENPARK` handles the delay of each modules as follows.

Our activity recognition engine suffers from around 30 second delay in finalizing the parking and leaving events. The delay can cause incorrect timing of GPS sample polling in check-in step, which in turn leads to inaccurate localization results. In our implementation, we compensate the delay by polling back 30 seconds for getting GPS samples from navigation software. The delay can also cause inaccurate "leaving now" information in check-out step. However, our observation is that drivers often take more than 30 seconds to factually pull out their car from a parking spot, which alleviate the negative impact of the delay caused by activity recognition.

5. ROBUSTNESS: ACTIVITY RECOGNITION BASED MALICIOUS USER CHECKING.

Suppressing bogus reports from malicious users is a key for securing the usefulness of crowdsourced parking information. A malicious contributor whose goal is solely to degenerate the quality of the SENPARK can submit a large amount of incorrect parking information to the system. Such dishonest contributors can greatly degrade the experience of SENPARK subscribers and adversely impact the viability of SENPARK.

We prevent malicious reports by using the activity recognition that is used for automated check-in and check-out procedures. The idea of our identification is that contributors with normal behavior tend to follow the expected motion state pattern between check-in and check-out events, while malicious users need to invest significant time and effort to pretend to be normal users. (Note that normal users are not on driving state between check-in and check-out events, but they should be on driving state after check-out event) Throughout our discussion, we simplify the robustness problem in the manner where users can be compromised but the phones are trustworthy. In other words, a malicious user does not manipulate GPS coordinates, accelerometer readings, and timestamps intentionally. There are many other studies focusing on handling device manipulations but they are out of the scope of the study.

Baseline schemes A few simple baseline mechanisms can be used as a first-level filter for preventing malicious contributors. First, the service provider can predetermine parking hotspots that are considered valuable to users and reject "leaving soon" information originating from other areas. Second, the service provider can rate limit the number of "leaving soon" messages per day. If a contributor generates more than the limit, the service provider can place the contributor in a blacklist. The advantage of these approaches is that both are easy to implement at the server side and require no additional information from the phone. However, these techniques cannot completely prevent malicious contributors — hotspot areas can be easily guessed by a malicious contributor, and rate limits typically need to be set conservatively and a malicious contributor could still provide considerable useless information once they figure out the limit.

ActCheck using activity recognition on the phone A second approach is to use activity recognition techniques on the phone to detect if the activities performed by the contributor is consistent with expectations, and it is called ActCheck in our system. ActCheck expects that a normal contributor should not in driving state between check-in and check-out events, and in driving state after check-out event. It is reasonable since by definition the car should be parked between check-in and check-out, and should be moving after check-out.

ActCheck consumes minimum energy since it only reads the results from activity recognition engine. No communication or manual operations from contributors are needed. Besides, our activity recognition engine only uses accelerometer sensor, which is very lightweighted in terms of power consumption.

The major problem of ActCheck is that malicious can manipulate their behavior to deceive the activity recognition engine. There are two major ways: 1) motorist in driving state pretending to be in walking state by driving very slowly and steadily, and 2) pedestrians pretending to be in driving state by vibrating and shaking phones to generate accelerometer noises that are common when driving. Our experience with activity recognition system indicates that such deception is very hard to achieve in practice — malicious contributors need to perform a perfect combination of extremely slow driving, usually less than 10 mph, throughout the time between check-in and check-out, or shaking the phone vigorously while walking to give the impression of driving. Both of them require malicious users with long time walking or tedious operations.

Another possible way to deceive our system is that a malicious user takes public transportation to a parking spot, walking out of our geofences, and return back to the parking spot and leaving with public transportation. Similarly, it requires a lot of time and effort for the malicious users to walk for a long distance and take public transportation tools.

Combining above approaches: While there are several approaches, each of them has pros and cons. The baseline schemes are the easiest to implement at the server side, and provide a first-order filter against malicious contributors. When the baseline schemes detect a pattern, for example, a contributor who send a large amount of parking information, ActCheck is triggered. ActCheck provides a second-order filter to detect if the activity pattern of the contributor follows expected behavior.

6. EVALUATION

In this section we evaluate the performance of SENPARK in two major aspects — the performance of leaving time estimation, and the performance of activity recognition based malicious user detection.

6.1 Performance of Leaving Time Estimation

We collected a walking trace dataset from five different participants at San Francisco downtown area. We specify five different parking spots, and allow them to have free activity of 20 minutes for each parking lot. We collect the GPS trace of the walking trajectory of each participants, and run our leaving time estimation algorithm against the trace dataset.

We measure the accuracy of leaving time estimation on its precision and recall, and specifically we are interested in how precision and recall change as we tune the

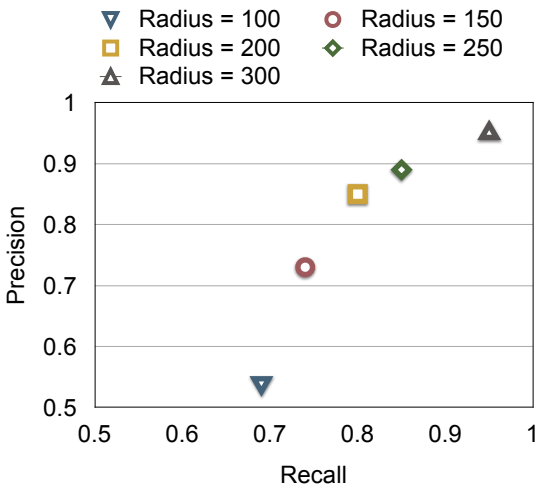


Figure 4: Leaving Time Estimation Performance when tuning geofence radius.

geofence radius settings. Here precision is defined as the ratio of correctly forecasted events (i.e., users have actually returned to the parking spots) out of all predictions, and recall is defined as the ratio of correctly forecasted events out of all true events where participants are back to parking spots. Thus precision means accuracy of forecasting while recall means sensitivity of forecasting in more general terms.

Figure 4 shows the precision and recall when the first circle radius varies from 100 meters to up to 300 meters, and the radius of the second circle is always 75 meters larger than the first circle. Since the third circle is placed for saving energy, we do not consider here. From this graph, we can conclude that our leaving time estimation algorithm is quite sensitive to geofence radius. When the radius is around 100 meters, both precision and recall are above 90%, but when the radius is increased to around 300 meters, the precision and recall are reduced to around 50%. The result matches to our life experience — when we are closer to a destination, the route and schedule become more predictable and less likely change than when we are far away from the destination. Around 90% precision and recall also give us enough confidence that our `SENPARK` service can provide reasonably accurate prediction of "leaving soon" information with around 2 minutes advance. However, if a service provider wants to put more weight on earlier forecasting than accurate forecasting, he has to find a larger radius in a trade-off relationship.

6.2 Performance of Malicious User Detection

Our second evaluation is about how accurately the activity recognition engine can detect malicious contributors. We focus on two types of malicious contributors, pedestrians and motorists. Throughout the remaining discussion, we shortly call the malicious user detection system (based on the activity recognition) ActCheck. We first evaluate the case where malicious contributors are unaware of ActCheck, and then evaluate the case where malicious contributors are aware of ActCheck and try to deceive it.

Malicious Users Unaware of ActCheck: We run ActCheck after contributors send either "leaving soon" message or "leaving now" message. For each case, we tune the running time of ActCheck from one minute to five minutes. Figure 5 shows the accuracy of ActCheck in classifying normal users and malicious users. From this figure, we find that the accuracy of ActCheck improves significantly as the running time increases. When ActCheck runs for five minutes, it can classify over 98% normal users correctly, while less than 5% malicious users can pass this classification filter. While the energy consumption of ActCheck increases with time, it is still energy-efficient since it only uses an accelerometer sensor [10].

Malicious Users Aware of ActCheck: In this experiment, several participants are recruited to act as malicious contributors to deceive ActCheck. Since ActCheck expects that a contributor behave as a pedestrian after sending "leaving soon" message and behaves as a motorist after sending "leaving now" message, we tried two malicious behaviors in this experiment: 1) act as a pedestrian in a moving vehicle, and 2) act as a motorist when walking.

To act as a pedestrian in a moving vehicle, users need to meet two conditions: acceleration (in x and y axes) and high frequency vibration (caused by moving vehicles) should be kept small enough. To keep the acceleration value small enough, users have to drive smoothly and slow. To force the high frequency vibration, users have to detach the phone from vibrating objects (i.e., vehicle and drivers); users can throw and catch the phone in vehicles. ActCheck can be deceived if both conditions are met. However, the effort to perform such deception becomes harder particularly when the running time of ActCheck is sufficiently long. To act as a motorist when walking is much harder, since it is difficult to mimic the high frequency vibration pattern of vehicles.

7. RELATED WORK

Recently many researchers have leveraged mobile sensing techniques for enhanced transportation experiences. Most research efforts are focused on handling the challenges during travelling time, including traffic congestion [7, 6], irregular public transit schedule [15], and unexpected road conditions [5]. Unfortunately not a serious attention has been paid on parking problem that not only affects the overall travelling experience but also

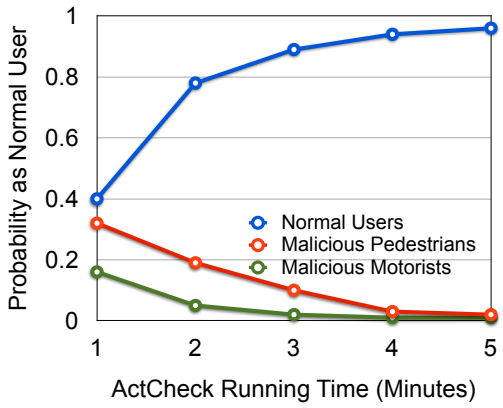


Figure 5: ActCheck accuracy vs. running time.

causes traffic problems. For instance, Shoup et al has observed that "about 30% of traffic is generated from cruising vehicles searching for parking spaces in downtown areas" [14]. Knowing available parking lots in advance can help drivers to best utilize various transportation modes as well as easing out stressful experiences.

An increasing number of mobile crowdsourcing applications allow users to share empty parking spot information. Examples include OpenSpot [8], Rodify [9], and many others. Among all these applications, Rodify is closest to us, as it allow users to share parking spots that are both available now and available soon. However, none of these applications solve the challenges we addressed in this paper, including inaccuracy, reducing manual operations, and detecting malicious users. Another relevant recent project is ParkNet [11], which installs ultra-sonic sensors on vehicles, and detects parking availability when vehicles drive by. This approach requires expensive infrastructure to be installed, and suffers from the same limitation as other approaches in that it only provides availability information and not when a spot is taken, unlike our approach.

There have been a spectrum of participatory sensing and personal sensing applications for mobile phones that use onboard sensors for various applications, such as obtaining images [3], activity patterns [10], predicting bus arrivals [15], and monitoring traffic and road quality [5, 6, 7]. Our work on SENPARK leverages mobile sensing for parking problem, and our novelty lies in the fact that we tailored activity recognition for detecting status switches and malicious users, and we also present trajectory analysis for leaving time estimation.

8. CONCLUSION

In this paper, we present SENPARK, which leverages smartphone users to contribute available-soon parking information accurately and with minimum manual operations. To increase reporting accuracy and reduce cumbersome manual operations, we propose a mobile system that predicts the leaving time of contributors such that available-soon information can be generated and shared with other drivers who need a parking spot. Furthermore, the presented activity recognition engine suppresses bogus reports from malicious users to secure the robustness and usefulness of crowdsourced parking sharing system. Our experimental results on real traced collected from San Francisco downtown area indicate that SENPARK provides over 85% prediction accuracy of leaving time with at least two minutes in advance.

Future directions: Our future research plans to expand SENPARK to address more real-world challenges in parking. First, we plan to integrate our parking information sharing system to navigation software suites to provide a better navigation experience that would guide you until your car is parked. Second, we will further study the data management issues in crowdsourced parking information sharing, such as how to incentivize participants, how to gather sufficient data, and how to store and query the crowdsourced parking information to provide efficient web service to consumers. Last but not the least, we are also interested in identifying potential privacy compromise in crowdsourced parking systems.

9. REFERENCES

[1] R. Arnott and E. Inci. An integrated model of downtown parking and traffic congestion. *Journal of Urban Economics*, 60:418–442, November 2006.
[2] N. Bilton. Finding That Prime Parking Spot With Primospot. http://bits.blogs.nytimes.com/2009/12/03/finding-that-prime-parking-spot-with-primospot/.
[3] N. Bulusu, C. Chou, and S. Kanhere. Participatory Sensing in Commerce: Using Mobile Camera Phones to Track Market Price Dispersion. In *UrbanSense*, 2008.
[4] J. V. Derbeken. Fatal stabbing over parking. http://articles.sfgate.com/2006-09-19/bay-area/17312921_1_parking-space-parking-spot-stabbed.
[5] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *ACM MobiSys*, 2008.
[6] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. Greengps: a participatory sensing fuel-efficient maps application. In *ACM MobiSys*, pages 151–164, 2010.
[7] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *ACM Mobisys*, 2008.
[8] J. Kincaid. Googles Open Spot Makes Parking A Breeze, Assuming Everyone Turns Into A Good Samaritan. http://techcrunch.com/2010/07/09/google-parking-open-spot/.
[9] N. Lamba. Social Media Tackles Traffic. http://www.wired.com/autopia/2010/12/ibm-thoughts-on-a-smarter-planet-8/.
[10] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *ACM Sensys*, November 3-5 2010.
[11] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe. Parknet: drive-by sensing of road-side parking statistics. In *ACM MobiSys*, 2010.
[12] U. D. of Transportation. Advanced parking management systems: A cross-cutting study. www.its.dot.gov/jpodocs/repts_te/14318_files/14318.pdf.
[13] SFMTA. SFPark - About the Project. http://sfpark.org/about-the-project/.

[14] D. Shoup. Cruising for parking. *Transport Policy*, 13(6):479–486, November 2006.
[15] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. In *ACM Sensys*, pages 85–98, 2010.