Unlocking the Smartphone's Senses for Smart City Parking

Jean-Gabriel Krieg[*][†], Gentian Jakllari[†], Hadrien Toma[*], André-Luc Beylot[†]
[*]ALSETT, [†]University of Toulouse – IRIT
{jgkrieg, htoma}@alsett.net, {jakllari, beylot}@enseeiht.fr

***Abstract*—Studies have shown that in places like New York City drivers often spend over $20\,\mathrm{min}$ looking for parking, contributing to as much as 30% of the total traffic. In response, cities like San Francisco have deployed systems capable of pointing drivers to the closest available parking spot. Unfortunately, such systems have gained little traction as they rely on specialized infrastructure that is expensive to build and maintain.**

We present SmartPark, a smartphone based system that relaxes the requirement for specialized infrastructure by relying on the smartphone's sensors and ubiquitous Wi-Fi and cellular infrastructure. To accomplish this, SmartPark addresses two major challenges, under the constraint of minimum impact on battery life: automatic transportation mode detection and location matching. Solved together, they enable SmartPark to automatically detect when a user pulls out of a parking spot, making it available again. It addresses each challenge using a combination of thorough statistical analysis of the sensor readings and a novel Random Forest based classification algorithm. Experimental results from 12 volunteers, using 7 different smartphones, in 3 different cities show that SmartPark can distinguish 9 different transportation modes with accuracy between $95.57 - 100\%$, enabling it to correctly detect unparking events virtually 100% of the time. This is accomplished with a minimum impact on battery life – running SmartPark on a fully charged LG Google Nexus 5 for $5\,\mathrm{h}$ straight caused the battery level to drop only about 4%.

I. Introduction

Someone named Bob is driving through downtown looking for on-street parking. Using an application on his smartphone he identifies a free spot at the next block, saving him valuable time. A Parsons Brickerhoff study [1] of New York City's Chinatown showed that on weekends 41% of the drivers spend more than 20 minutes looking for on-street parking. This figure increases to 54% on weekdays. Using this application also reduces Bob's carbon footprint; in congested urban areas 30% of the traffic is due to drivers looking for parking [2] [3]. Once parked, Bob leaves the vehicle and goes about his business. Upon his returning to the vehicle and pulling out of the parking spot, the application computes automatically the time spent parking and charges Bob's account. This saves Bob the hassle of having to anticipate how long he would be away, go to the ticket machine to pay and place the receipt on the dashboard. At the same time it saves the city a significant amount of money. Cincinnati, for example, is owed about $12 million from unpaid parking tickets dating back to 2005 [4]. Furthermore, as the application relies on the smartphone's sensors and infrastructure already available, cities will not have to invest in building and maintaining an extensive parking payment infrastructure. Finally, the application has a negligible effect on the phone battery.

Unfortunately, such application does not exist. The major scientific obstacle being that the problem of automatically identifying when a user returns to her vehicle and pulls out of the parking spot – the return-to-vehicle problem – without draining the phone's battery remains open. One might be tempted to think that users should just volunteer this information. However, given that the payment amount depends on it such system would be vulnerable to abuse. Indeed, even when parking is free, the Google Open Spot experiment has shown that most users neglect to inform the system when they pull out of their spot [5]. GPS-based solution to the return-to-vehicle problem have been proposed [6], [7], [8] and shown to be highly accurate. Unfortunately, relying on GPS can quickly drain the battery, a non-starter for most smartphone users. ParkSense [2] proposes a solution using only the smartphone's wireless interface so as to reduce the energy consumption but to do so it relaxes the requirement for accuracy. We believe that for a parking payment infrastructure, with all the financial implications it entails, to win wide adaptation it will have to be both accurate and energy efficient.

We present SmartPark, the first system that solves the return-to-vehicle challenge with accuracy reaching 100% while having a negligible impact on battery life. SmartPark addresses the challenge in two steps: One, it solves the problem of automatic transportation mode detection. The solution to this problem enables SmartPark to identify if a user has returned to the same kind of vehicle as the one she parked. Two, it solves the problem of identifying a particular location. The solution to this problem enables SmartPark to identify if a user has returned to the vehicle she parked instead of just another vehicle of the same type. To solve these problems, SmartPark relies on four innovations. First, unlike traditional approaches that rely on a single sensor, such as the wireless interface or accelerometer, SmartPark leverages all sensors available on modern smartphones, including the luminosity and atmospheric pressure sensors, to detect the transportation mode. Sound, light and pressure readings inside a subway, for example, may be different from the readings when riding a motorbike, allowing to distinguish these two transportation modes. Second, SmartPark computes an exhaustive number of summary statistics for every sensor reading, leading to 2880 explanatory variables. Third, SmartPark uses a Random Forrest [9] based approach for classifying the sensor readings, in real time, into one of the

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

transportation modes one is likely to use in a city: walking, bicycle, bus, car, subway, motorbike, train, tram, airplane. Fourth, SmartPark introduces a novel similarity coefficient for matching two physical locations using Wi-Fi and cellular base station signals.

In summary, our contributions can be summarized as follows:

- We introduce the first system capable of detecting virtually every mode of transportation found in modern cities with accuracy approaching 100% while relying exclusively on the sensors available on modern smartphones.
- We introduce an algorithm for location matching using readings from Wi-Fi and cellular base stations.
- Combining the two above, we introduce SmartPark, a smartphone based system that solves the return-to-vehicle challenge with accuracy approaching 100% while having a negligible impact on battery life.
- We implement SmartPark and test it on 7 different smartphones with the help of 12 volunteers from 3 different cities. Over 30 hours of experiments show that: a) SmartPark correctly detects a return-to-vehicle (unparking event) in 29 of 29 scenarios without any false positives, and b) SmartPark has a minimal impact on the battery consumption. Running SmartPark on a fully charged LG Google Nexus 5 for $5\,\mathrm{h}$ straight caused only about 4% drop in battery charge.

The remainder of the paper is organized as follows. Section II describes the SmartPark architecture while Section III describes the transportation mode detection algorithm. In Section IV, we introduce the mechanism for identifying a specific location. In Section V, we present the performance evaluation. Finally, Section VII concludes the paper.

II. System Overview

SmartPark enables the automatic detection of a user pulling out of a parking spot by leveraging the smartphone's sensors and ubiquitous Wi-Fi and cellular infrastructure. At the heart of SmartPark is the ability to mine readings from seven different sensors so as to identify virtually any transportation mode available in modern cities. The large number of sensor readings is necessary for overcoming the inherent erroneous nature of the inexpensive sensors with which modern smartphones are equipped. Once the transportation mode is detected, SmartPark can tell if the user is back in the *same* kind of vehicle as when she parked. With a similarity matching on the Wi-Fi and cellular signals received, SmartPark can finally tell if the user is back in *her* vehicle and starting to pull out of the parking spot and not just any similar vehicle. Figure 1 shows the architecture of SmartPark. SmartPark can be in one of the 4 following states: 1) The user is looking for parking: SmartPark detects automatically the user's vehicle type and sends a request to the SmartPark server via the Wi-Fi or cellular networks asking for the available parking spots in the vicinity. 2) The user parks: SmartPark scans the available Wi-Fi and cellular base stations available to create a location profile and saves it for future use. 3) The vehicle is parked: the user can go about her business, walking around, taking the bus, subway, tram or any other transportation mode. SmartPark, in the mean time, triggers a transportation mode detection every $10\,\mathrm{s}$. 4) The user is unparking: SmartPhone detects this event because a. the user is using the same transportation mode as when she parked, and b. she is back at the same location. Once the user unparks, SmartPark sends a message to the SmartPark server notifying of the user's leaving the parking spot.

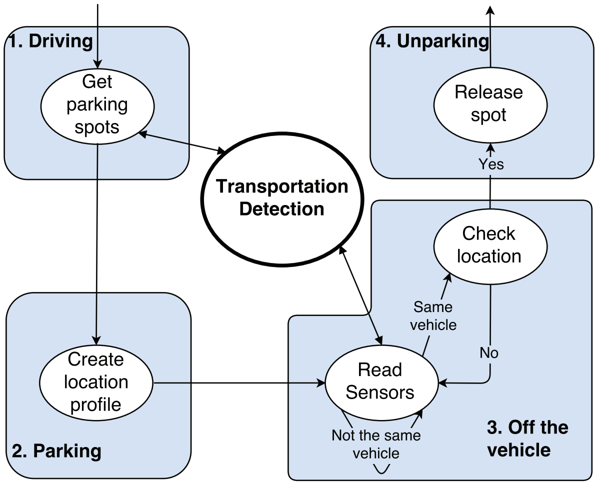

Figure 1. **SmartPark Architecture**. The database of the available parking spots is maintained by a central server. When a user parks, SmartPark creates a profile of the location using readings from Wi-fi and cellular base stations, and of the vehicle using the Transportation Mode detection module. Once the user parks, SmartPark checks periodically the transportation mode so as to detect when she is using again the same kind of vehicle. If that is the case, SmartPhone runs a location matching and, if positive and the users start pulling away in the vehicle, it knows the user has unparked and notifies the server that the spot is available again.

SmartPhone relies on two major functionalities:

1) Transportation mode detection.
2) Location matching.

We describe each in detail in the next sections.

III. Transportation Mode Detection

In this section, we present SmartPark's module for identifying the transportation mode using a smartphone. Needing only $2\,\mathrm{s}$ of sensor readings, it can decide if a user is using one of the following transportation modes: **walking, bicycle, bus, car, subway, motorbike, train, tram, airplane**. This unprecedented granularity, speed and accuracy is achieved thanks to two innovations. One, while previous approaches rely on a single sensor, usually the accelerometer [2], SmartPark leverages a combination of sensor readings, including sound, luminosity and pressure levels, to uniquely profile every transportation mode. Two, an innovative classification algorithm based on Random Forest [9] for matching readings to a transportation mode in real time.

A. Leveraging Multiple Sensors

A key insight underlying SmartPark is that all sensors with which modern smartphones are equipped can provide useful information as to what a particular user is doing. Sound and light levels while riding a motorbike can be very different

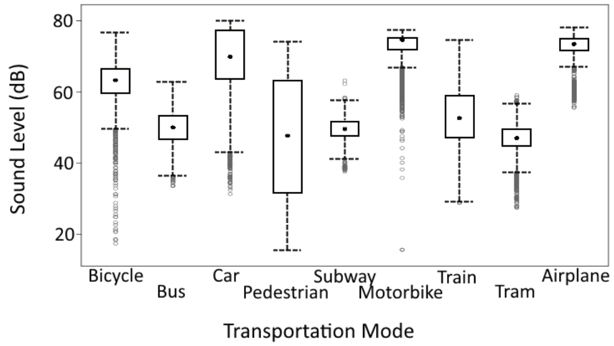

Figure 2. Boxplot of sound level readings for the 9 transportation modes. The motorbike readings show the highest median value and a tight box. However, the results make it clear that it is impossible to detect all transportation modes using only this sensor.

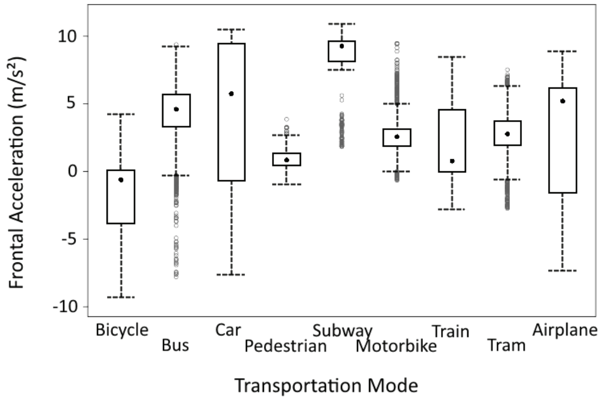

Figure 3. Frontal acceleration readings for the 9 transportation modes. The readings in the subway show the highest median value and a tight box, consistent with the acceleration one experiences at every stop of a subway. However, as with the sound, these readings alone cannot enable reliable transportation mode detection.

from those inside a car or a bus. Pressure levels in a subway usually running underground may be different from those inside a tram. At the same time, these are inexpensive and inherently erroneous sensors and cannot be the lone source for transportation detection. Fig. 2 shows a boxplot of sound levels measured using a LG Nexus 5 while using the 9 transportation modes SmartPark supports. The measurements show that sound sensor readings might help identify certain transportation modes. For example, the readings while riding a motorbike have the highest median value and a tight box. However, several activities produce sound readings too similar to be used to distinguish one from the other. A similar phenomena is observed with the frontal acceleration readings show in Fig. 3. Here it is the subway producing the highest median value and a tight box – consistent with the acceleration one experiences repeatedly at every station. Nonetheless, measurements show significant overlaps in frontal acceleration readings across the car, tram and airplane.

To overcome the limitations of using a single sensor, SmartPark finds a way to combine readings from the entirety of the sensors available on smartphones. As Fig. 4 shows,

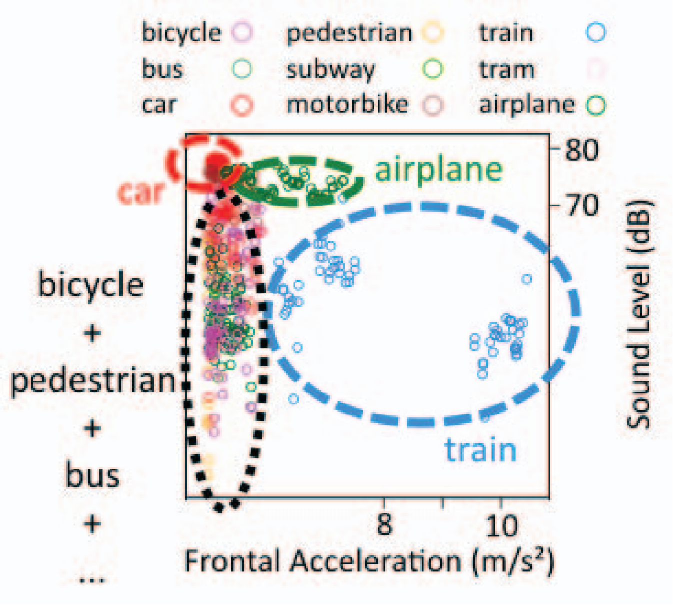

Figure 4. Sound and acceleration readings projected on a pairs plot: max of frontal acceleration and mean of sound. Combining just two sensor readings allows the detection of three transportation modes: car, airplane and train. However, it is impossible to distinguish the other transportation modes, explaining why SmartPark uses more than two sensors.

using multiple sensor readings can significantly improve our capability of distinguishing different transportation modes on a smartphone. Jointly using readings from only two sensors - the frontal acceleration and sound levels - one can already detect three transportation modes: car, airplane and train – something we could not do using only one sensor.

Therefore, SmartPark uses all the sensors available on modern smartphone to detect the transportation mode. Specifically:

- Accelerometer (three axes).
- Gyroscope (three axes).
- Orientation (three axes).
- Magnetic field sensor (three axes).
- Luminosity sensor.
- Atmospheric pressure sensor.
- Microphone.

B. Summary Statistics

Table I shows the summary statistics we compute for the temporal, frequency and wavelet transform of every sensor reading. We have erred on the side of being exhaustive and selected the majority of the commonly used summary statistics for a total of 64. In Section III-C, we present an algorithm that automatically selects the statistics best at identifying the 9 transportation modes SmartPark supports. Note that, this computation is done once offline and the results are included in SmartPark. At run time, SmartPark simply collects sensor readings and uses the decision trees, as shown in Section III-C, to make a decision as to the transportation mode.

C. Classification

We collect 15 sensor readings for which we calculate the FFT and the wavelet transform. For each of the three versions

Domains	Temporal, Frequential, Wavelets
Location statistics	Mean, Median, Harmonic & geometric mean, RMS
Spread statistics	STD, Var,1st-Q, 3rd-Q
Shape statistics	Kurtosis, Skewness, Entropy
Integral and derivative	Surface, growth rate
5 Peaks and 5 Valleys	Value, magnitude, FHWM, 2 inter-distances

Table I
STATISTICS EXTRACTED FROM EACH SENSOR TO CREATE THE MODEL VARIABLES.

of the signal – temporal, frequency and wavelet transform – we compute the 64 summary statistics, as shown in Table I. That gives us a total of $15 \times 3 \times 64 = 2880$ *explanatory variables*. The challenge we address in this section is using this significant amount of information for classifying sensor readings into one the 9 transportation modes, the target variables.

Several classification models have been proposed in literature with the Partial Least Squares (PLS) [10] (itself a major evolution of Principal Component Analysis (PCA)) one of the more popular. It tries to find a linear regression model by creating a new plane from the explanatory and target variables [11]. PLS components are selected so that their covariance with the target variable is maximum. Thus from a list of predictors, PLS extracts the linear combinations of the predictors - also called latent factors - to classify the target variables. PLS regression is particularly efficient in cases such as ours where the explanatory variable space is significantly bigger than the target variable space. However, as PLS is based on a linear approach, it needs to compute many linear combinations of the explanatory variables so as to predict one or more transportation modes [12]. Given the variety of explanatory variables SmartPark uses, this would result in a very high number of computations making it a poor choice for a smartphone based solution. Therefore, we opted for tree-based classifiers.

Classification trees are hierarchical structures separating experiments from each transportation mode according to splitting rules on the observed variables. At each node, the algorithm finds the explanatory variable and the threshold for this variable that partitions the observations into two branches such that the disorder in each branch is minimized. In this context, disorder is defined as having few explanatory variables per target variable. To quantify the disorder we use the entropy, calculated at each node, i, as follows:

$$D_i = - \sum_{m \in \forall modes} \frac{n_{m,i}}{N_i} log(\frac{n_{m,i}}{N_i})$$

where N_i is the number of individuals in the i-th node and $n_{m,i}$ the number of individuals corresponding to the m-th mode in the i-th node. Finally, the tree leaves correspond to the most likely transportation modes, as shown in Fig. 5.

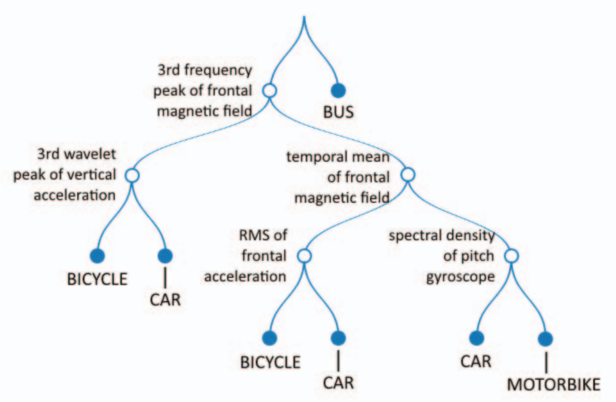
3rd frequency peak of frontal magnetic field
BUS
3rd wavelet peak of vertical acceleration
temporal mean of frontal magnetic field
BICYCLE
CAR
RMS of frontal acceleration
spectral density of pitch gyroscope
BICYCLE
CAR
CAR
MOTORBIKE

Figure 5. One ending branch of one of the trees computed by SmartPark RF model.

While a single tree can work well, it is not optimal due to the potential correlation between the large number of explanatory variables SmartPark uses. To minimize the effects of correlation, we build a Random Forest (RF) [9]. It consists of selecting the best explanatory variable at every branch of the tree by considering only a randomly selected subset of the total of the explanatory variables. We use this approach and compute 500 such trees offline. The trees are loaded into SmartPark which simply needs to “browse" them when trying to detect a transportation mode – a step well within the computation capabilities of modern smartphones.

IV. LOCATION MATCHING: IS THIS **MY** VEHICLE ?

When a user parks her vehicle (car/motorbike/bicycle), she enters the parking spot number in the SmartPark application. Automatically, SmartPark creates a *location profile* by leveraging the ubiquitous Wi-Fi and cellular infrastructure. In particular, the location profile is computed as follows:

$$\begin{aligned}
\mathbf{S_1} &= (S_1(1), S_1(2), ..., S_1(a)) \\
\mathbf{Ws_1} &= (Ws_1(1), Ws_1(2), ..., Ws_1(a)) \\
\mathbf{C_1} &= (C_1(1), C_1(2), ..., C_1(b)) \\
\mathbf{Wc_1} &= (Wc_1(1), Wc_1(2), ..., Wc_1(b))
\end{aligned}$$

where $\mathbf{S_1}$ represents the SSIDs of the a access points, $\mathbf{Ws_1}$ the RSSIs of the corresponding APs, $\mathbf{C_1}$ the Cell ID of the b stations and $\mathbf{Wc_1}$ the RSSIs of the corresponding stations.

When SmartPark detects that the user is using the same transportation mode as the one she did when she parked, it checks whether the location is also the same[1]. For this, it computes a profile of the current location and compares it to the one stored at the time of parking. The two profiles are compared and a similarity coefficient, C, is computed as follows:

$$\begin{aligned}
C = \quad & \frac{1}{\sum_{i=1}^{a} Ws_1(i)} \sum_{i=1}^{a} Ws_1(i)(1 - |\frac{Ws_1(i) - Ws_2(i)}{Ws_1(i)}|) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (1) \\
+ \quad & \frac{1}{\sum_{i=1}^{b} Wc_1(i)} \sum_{i=1}^{b} Wc_1(i)(1 - |\frac{Wc_1(i) - Wc_2(i)}{Wc_1(i)}|)
\end{aligned}$$

[1] After a user parks her car, for example, there is a possibility that she meets a friend who offers her a ride in her car.

where $Ws_2(i)$ and $Wc_2(i)$ represent the RSSIs of the second scan corresponding to APs $S_1(i)$ and $C_1(i)$ - 0 if not found.

In selecting Eq.(1) we were driven by three principles:

- *Proportion of APs found*: we take into account the proportion of APs in common between the two location profiles by assigning "0" to $Ws_2(i)$ for any mismatch.
- *Compare raw RSSIs values*: instead of a binary model, we compare the RSSI values of the two respective profiles. The smaller the difference, the higher the similarity coefficient.
- *Favor the highest RSSI values*: the main novelty of our coefficient is that we add a weight to favor the APs/BSs with the highest RSSI values. The idea being that APs/BSs with stronger RSSI are more likely to be closer to the actual parking location.

V. Performance Evaluation

In this section, we evaluate SmartPark in terms of accuracy at detecting the transportation mode, identifying a specific vehicle, energy consumption and its overall capability of identifying when a user parks/unparks her vehicle. All the data are from real experiments carried in 3 different cities, using 7 different smartphones and 12 volunteers.

A. Classification Accuracy

Method: Twelve volunteers are given one of the 7 smart devices, Sony Xperia E3, LG Google Nexus 5, LG Google Nexus 6, LG G2, Huawei Honor 4X, Samsung Galaxy S5, Samsung Galaxy S3 mini, and are asked to enter regularly their transportation mode as they go about their normal daily lives. SmartPark records the 15 sensor readings, as described in Section III, with the highest frequency rate available, often 200Hz, computes the explanatory variables and makes its own decisions about the transportation mode using the 500 pre-computed classification trees[2]. In the end, we collect data from about 28 hours of experiments during which SmartPark makes 10282 decisions as to the transportation mode. To measure the accuracy we compare what the user entered with the transportation mode computed by SmartPark. Furthermore, to exclude the element of chance from the results we compute Cohen's kappa criteria:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

where p_o represents the observed agreement between real transportation mode and predicted transportation mode and p_e represents the probability of randomly correct detection.

Results: At first we investigate whether choosing the Random Forest approach over PLS or a single tree was the right decision. **Table II** shows that indeed Random Forest, as implemented by SmartPark, is by far the most accurate approach, almost doubling the accuracy of a single tree and PLS.

Method	Accuracy	Kappa
PLS	0.69	0.67
Single Tree	0.54	0.53
Random Forest	0.99	0.98

Table II
Random Forest with 500 trees achieves 98.72% accuracy in transportation mode detection.

Next, we evaluate the accuracy of SmartPark at detecting each of the 9 transportation modes it supports. **Table III** shows the confusion matrix of the 10282 transportation mode decisions SmartPark took. The data shows SmartPark detects every transportation mode with accuracy that never drops below 95%. Equally important, the number of false positives is negligible.

Finally, we compare SmartPark to 5 different approaches from the literature, namely Reddy [13], Stenneth [6], Zheng [8], Wang [14] and Peaks [15]. **Table IV** shows that: One, SmartPark is by far the most accurate. Two, SmartPark detects by far the most modes of transportation. The closest in terms of modes of transportation supported are Peaks and Wang which support 5 to SmartPark's 9.

B. Identifying Own Vehicle

Method: A volunteer is asked to perform a large number of parking and unparking events with her vehicle in the course of several days while using SmartPark. Then she is asked to perform a large number of what we call “false" parking/unparking events: she parks her vehicle, goes about her business then takes a ride back in a friend's vehicle. To create the ground truth, she inputs on her phone the kind of event (parking or unparking) she performs every time. By the end of the experiment, the user performed 29 real parking/unparking and 27 false parking events.

To quantify the accuracy, we compare what the user declared with what SmartPark computed. We compare the accuracy of SmartPark to that of ParkSense (the two versions proposed in [2]), the traditional and weighted version of Tanimoto index [16] and the *Jaccard Index*. For two locations, the Jaccard Index is computed as $J = \frac{|\mathbf{S_1} \cap \mathbf{S_2}|}{|\mathbf{S_1} \cup \mathbf{S_2}|}$.

Results: Figure 6 shows the results for different values of the similarity threshold (Eq.(1), Section IV). The data shows that the traditional Jaccard index is too conservative, leading to its missing many real parking/unparking events. Tanimoto and ParkSense perform much better, however, even for similarity coefficient 0.5, for which all approaches perform best, they suffer from a large number of false positives. This is due to the fact that they do not use RSSI information. SmartPark, on the other hand, identifies 97% of the real parking/unparking events and triggers 0 false positives.

C. Energy Consumption

Figure 7 shows what happens to the battery level when a user starts with a full charge and uses SmartPark. For comparison, we also include the effect of using the GPS and Wi-Fi continuously. The data shows that SmartPark has a

[2] The 500 trees are pre-computed using a different set of experiments. This is a one time computation and the results are entered into SmartPhone which uses them as longs as it is in use.

Transport	Bicycle	Bus	Car	Subway	Motorbike	Pedestrian	Train	Tram	Airplane	Accuracy
Bicycle	802	0	2	0	0	2	0	0	0	99.50%
Bus	0	837	0	0	0	0	0	47	0	97.51%
Car	3	8	1747	0	0	6	0	0	0	99.04%
Subway	0	1	0	583	0	0	0	26	0	95.57%
Motorbike	3	0	7	0	835	7	0	0	0	98.00%
Pedestrian	0	0	0	0	0	973	0	0	0	100.00%
Train	0	0	0	0	0	0	830	0	0	100.00%
Tram	0	7	0	0	0	0	0	1506	0	99.54%
Airplane	0	7	0	0	0	0	0	6	1037	98.76%

Table III
CONFUSION MATRIX OF SMARTPARK FINAL MODEL.

Mode	SmartPark	Peaks	Wang	Reddy	Zheng	Stenneth
Bicycle	99%	-	81%	87%	66%	89%
Bus	98%	78%	58%	-	67%	88%
Car	99%	-	72%	-	86%	88%
Subway	96%	65%	53%	-	-	-
Motorbike	98%	-	-	-	-	-
Pedestrian	100%	95%	81%	95%	89%	98%
Train	100%	68%	-	-	-	98%
Tram	99%	84%	-	-	-	-
Airplane	99%	-	-	-	-	-
Motorized	99%	74%	54%	95%	76%	91%
Latency	2s	4s	8s	1s	2s	-

Table IV
SMARTPARK SYSTEM ALLOWS TO LARGELY INCREASE THE PERFORMANCE OF CURRENT TRANSPORTATION DETECTION SYSTEM, WITHOUT INVOLVING A HIGH LATENCY.

negligible effect on the battery and significantly lower than using just the Wi-Fi for example.

To elucidate why SmartPark consumes so little energy, we analyze the energy needs of its two core functionalities: detecting the transportation mode and location matching.

Transportation Mode Detection: SmartPark uses $2\,\mathrm{s}$ of sensor readings for detecting the transportation mode with the process repeating every $10\,\mathrm{s}$ when the car is parked. We measured the energy consumed by the 6 sensors SmartPark exploits and the energy required to calculate the summary statistics on this $2\,\mathrm{s}$ sample. We found it to be around $197\,\mathrm{mA}$ for the 7 smartphones used in our experiment. This represents about $1615\,\mathrm{mJ}$ every $10\,\mathrm{s}$.

Location Check: We measured the energy consumption of the Wi-Fi and network passive scans on the 7 smartphones used in this study and found it to be around $117\,\mathrm{mA}$ for a period of $1.07\,\mathrm{s}$, representing about $513\,\mathrm{mJ}$ per location check.

D. *Overall System Performance*

We ask 7 volunteers from 3 different cities to use SmartPark for a period of a few months; the only action asked of them is to signal on their phones whenever they park and unpark their vehicles so as to create the ground truth. Five of them are regular car users while the other 2 usually bike.

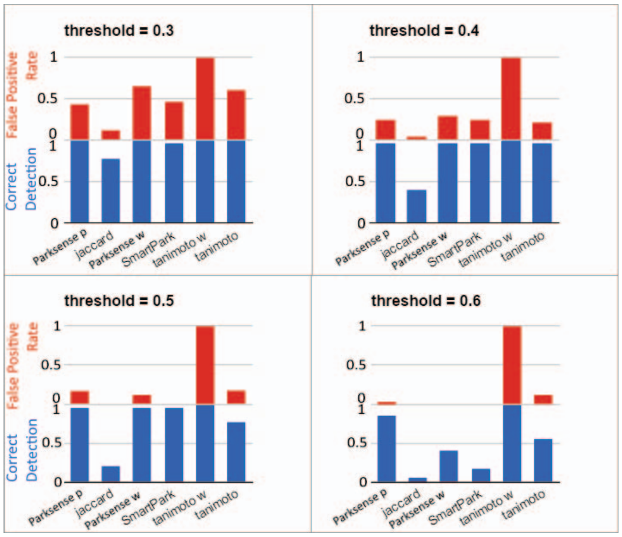

Figure 6. Correct Detection proportion (bottom blue) and False Alarms proportion (top red). SmartPark, using a similarity threshold of 0.5, has 0 false positives over the 27 false unparking events while achieving 97% correct detection over the 29 real unparking events.

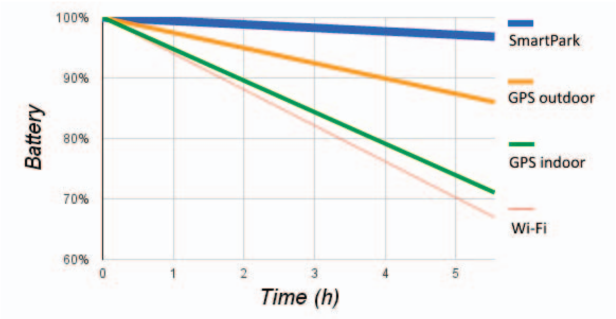

Figure 7. After $5\,\mathrm{h}$ of using SmartPark, the battery level has dropped only 4%.

At the end of the experiment, the data showed that the volunteers signaled 27 parking and unparking events while using a car and 2 parking and unparking events while using a bicycle. SmartPark managed to correctly detect the parking and unparking 100% of the time, with zero false positives.

VI. Related work

Several works have proposed using the various sensors available on smartphones to identify user activities. Bao et al [17] proposed to classify user activities using the accelerometer. In [18] the authors developped a multimodal sensors board to detect various human activities such as walking, brushing the teeth, sitting, etc. However, most sensors proposed are currently present on smartphones, making their board obsolete. GPS has been considered for detecting transportation mode [8], however, it has been shown to be inaccurate and energy hungry. In [6] the authors proposed adding GIS information to improve the accuracy of transportation mode detection whereas [14] introduced the accelerometer values. [13] and [7] propose to combine accelerometer and GPS and achieve 94% accuracy in classifying motions from motorized vehicles, pedestrian activities and bicycling. A solution based on GSM and Wi-Fi is proposed in [19] and is shown to be around 85% accurate. However it relies on an infrastructure and thus needs recalibration with each minor change.

Most new indoor parking lots install custom infrastructure allowing users to find free parking spots. Having the same kind of system for their downtowns has become a major challenge for cities worldwide [20], [21], [2]. San Francisco, for example, has installed custom wireless nodes for facilitating on-street parking [22]. However, custom infrastructures can be expensive to install and maintain. SmartPark on the other hand relies exclusively on the smartphones users already own. One solution proposed to relax the requirement for custom infrastructure is the crowdsourcing [23]. However, as all crowdsourcing based solutions, it suffers from a chicken-egg problem: the quality of the solution hinges on user adaptation while the user adaptation hinges on the quality of the solution.

VII. Conclusions

We presented SmartPark, a system that relies exclusively on the smartphone sensors to automatically point drivers to the nearest parking spot and bill them for the amount of time parked. To accomplish this, SmartPark solves the return-to-vehicle problem by solving the automatic transportation mode detection and location matching problems. The solution is based on an exhaustive statistical analysis of the sensor readings, generating 2880 explanatory variables, and a Random Forest based approach for classifying the explanatory variables into one of the 9 supported transportation modes in real time on a smartphone. We evaluated SmartPark on 7 different smartphones, in 3 different cities with the help of 12 volunteers and showed over 95% accuracy in transportation mode detection and in return-to-vehicle scenarios.

As future work, we plan to negotiate with city halls that may be interested in adapting SmartPark to handle parking management for a particular area in experimental form or city wide.

References

[1] Urbitran/Parsons Brinckerhoff, Eng-Wong, Taub & Associates, Howard/Stein-Hudson Associates, Inc., HOK Planning, "Canal Area Transportation Study Track II," New York Metropolitan Transportation Council. New York, NY, 2008.

[2] S. Nawaz, C. Efstratiou, and C. Mascolo, "ParkSense: A Smartphone Based Sensing System for On-street Parking," in *ACM MobiCom '13*, 2013, pp. 75–86.

[3] D. C. Shoup, "Cruising for parking," *Transport Policy*, vol. 13, no. 6, pp. 479–486, 2006.

[4] J. Hanselman, "Cincinnati has lots of unpaid parking tickets," *WVWU*, 2014.

[5] Google open spot: A useful application that no one uses. [Online]. Available: http://www.androidauthority.com/googlelabs-open-spot-a-useful-application-that-noone-uses-15186/

[6] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and gis information," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, vol. 37. ACM, 2011, pp. 54–63.

[7] H. J. T. Tao Feng, "Transportation mode recognition using gps and accelerometer data," in *Transportation Research Part C: Emerging Technologies*. ACM, 2013, pp. 118–130.

[8] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on gps data," in *ACM UbiComp '08*.

[9] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[10] W. H, "Partial least squares," *Encyclopedia of Statistical Sciences*, 2006.

[11] R. D. Tobias, "An introduction to partial least squares regression," 1995.

[12] I. Cramer, Richard D., "Partial least squares (pls): Its strengths and limitations," *Perspectives in Drug Discovery and Design*, vol. 1, no. 2, pp. 269–278, 1993.

[13] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Trans. Sen. Netw.*, vol. 6, no. 2, pp. 13:1–13:27, 2010.

[14] S. Wang, C. Chen, and J. Ma, "Accelerometer based transportation mode recognition on mobile phones," in *Proceedings of the 2010 Asia-Pacific Conference on Wearable Computing Systems*, ser. APWCS '10.

[15] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *ACM SenSys '13*.

[16] M. Kjaergaard, M. Wirz, D. Roggen, and G. Troster, "Mobile sensing of pedestrian flocks in indoor environments using wifi signals," in *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, March 2012, pp. 95–102.

[17] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, 2004, vol. 3001, pp. 1–17.

[18] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Proceedings of the 4th International Conference on Pervasive Computing*, ser. PERVASIVE'06, 2006, pp. 1–16.

[19] J. B. M. Mun, D. Estrin and M. Hansen, "Parsimonious mobility classification using gsm and wifi traces," in *ACM SenSys '08*.

[20] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," in *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on*, 2008.

[21] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Intelligent parking space detection system based on image processing," *International Journal of Innovation, Management and Technology*, vol. 3, no. 3, 2012.

[22] SFpark. (2015). [Online]. Available: http://www.sfpark.org/

[23] T. Yan, B. Hoh, D. Ganesan, K. Tracton, T. Iwuchukwu, and J. sik Lee, "Crowdpark: A crowdsourcing-based parking reservation system for mobile phones," 2011.