Mobile Networks and Applications (2019) 24:171–183
https://doi.org/10.1007/s11036-017-0987-6

SMinder: Detect a Left-behind Phone using Sensor-based Context Awareness

Haibo Ye[1] · **Kai Dong**[2] · **Tao Gu**[3] · **Zhiqiu Huang**[1]

Published online: 26 December 2017

Abstract

Forget your smartphone in the car again? This happens often in our daily lives, sometimes even makes troubles. In this paper, we present SMinder, an effective, low power approach to remind user take the phone when getting off the car. Based on the context awareness techniques in mobile sensing, we detect the situation of forgetting to take the phone when getting off the car. SMinder requires neither any infrastructure nor any human intervention. It only uses low power smartphone sensors. Namely, the smartphone detects by itself whether it is left behind and remind the user before he leaves the car. SMinder reminds the user with high accuracy and minimum energy consumption, making it realistic for real-world use. Compared to the existing approaches, SMinder is cheaper and easier to use. Our experiments with the prototype system demonstrate the performance, scalability, and robustness of SMinder.

Keywords Smartphone sensing · Left-behind phone · Context detection · Context inferring

1 Introduction

This paper describes evaluations and experiences with SMinder, a novel system to detect a left-behind phone and remind the user when getting off the car. The motivation comes from the observation that people use smartphones more in cars today and often forget to take the phone when getting off the car, which sometimes cause troubles. Imagine that when you enter the office/apartment and comfortably sitting in the chair, suddenly realize the phone is left behind in the car, that will be very annoying. Reports also show that if you don't get the phone back in time, it may be damaged by the high temperature at summer days[1] and has a greater risk of being stolen. The situation becomes more serious when you leave the phone in a taxi. Getting back the forgotten phone is one of the topest accessed services in the Taxi system and the car sharing platforms such as Uber[2] in China, and only a few of them can get the phone back, since you cannot know the phone is taken by the driver or the next passenger. Nowadays, smartphone is a privacy-sensitive device [4], the privacy disclosure is more serious than the economic loss caused by a lost phone.

To assess the need for SMinder, we conducted a user survey using mtc.baidu.com, between October 1th and December 15th 2016, asking the participants about their views regarding leaving smartphone in cars. The survey was terminated upon receiving 1000 valid responses. The purpose was to answer questions such as: How often do the drives forget their smartphones in the car? Do you need a tool to remind you to take the phone when getting off the car? How much smartphone power consumption is acceptable for you to have this tool running on your phone? The answers, which will be detailed later, confirm

✉ Haibo Ye
yhb@nuaa.edu.cn

Kai Dong
dk@seu.edu.cn

Tao Gu
tao.gu@rmit.edu.au

Zhiqiu Huang
zqhuang@nuaa.edu.cn

1 College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, People's Republic of China

2 School of Computer Science and Engineering, Southeast University, Nanjing, 211189, People's Republic of China

3 School of Computer Science and IT, RMIT University, Melbourne, Australia

[1] The temperature in car can reach 60 °C or more under the sun in summer.

[2] Uber help service: https://help.uber.com.

that drivers/passagers often forget their smartphones in cars, and users mostly need a tool which can remind them to take the phones when getting off the car, and about 5% of the smartphone energy consumption is acceptable to let the application running on the phone.

In order to remind the user when getting off the car, an intuitive solution is to use the smartphone ant-lost techniques, which checks whether your phone is nearby all the time. Approaches based on Bluetooth [2, 9] and RFID [6, 14] have been proposed. Chavan [2] designed a Bluetooth based approach. In this approach, the user needs to wear an iBeacon [9] on body, which is powered by a button battery and can continue to work for nearly six months theoretically. The smartphone periodically scans the Bluetooth signal strength from the beacon, and translates it to distance. If the distance is above a threshold, the smartphone will remind the user. Such methods cannot avoid false detections caused by the unstable Bluetooth signal strength, and the beacon needs to be on body all the time, which is not cheap and easy to use. The US patent 9070276 [8] deals with the problem using the car information system. It reminds user based on engine and door state information. The approach performs well but has strong assumptions on the car and the smartphone. The smartphone needs to connect and transfer sensor data to the car, which is feasible for a driver but not for passengers. The connection process is complicated and has security risks.

In this paper, we present a novel, effective system SMinder. The general intuition is the smartphone can detect its surroundings by the smartphone sensors [11]. SMinder makes use of the light, accelerometer, magnetic, and barometer sensors of the smartphone only. Based on context awareness and reasoning, after dealing with the challenges such as noisy sensor readings and complex environmental changes, we can detect the situation when the user is getting off while the smartphone is still in the car, and timely remind the user. Briefly, in most of the time, SMinder is running in a low power mode, where it only periodically collects the barometer samples to detect whether the user is in a running car.[3] Once SMinder find out that the user is in a running car, it will switch to the running mode. SMinder uses the accelerometer sensor to detect whether the car is stopped, use the light sensor to detect whether the smartphone is in pocket or is placed in the car, and use the magnetic sensor to detect whether the car door opens or closes. Based on the detected context of the car and the context inferring rules, we design a reminder decision mechanism to decide when to remind the user.

In summary, we make the following contributions:

1. We present a novel system to detect a left-behind phone and remind the user when getting off the car. SMinder only uses the smartphone sensors and does not require any infrastructure support.
2. we conducted a user survey about their views regarding leaving smartphones in cars. The result shows that drivers/passagers often forget their smartphones in cars and they need a remind tool.
3. We design several novel techniques to detect the contexts in the car. For example, the way to detect the car door open by magnetic signatures is bright new.
4. We conduct extensive field studies to evaluate the performance of SMinder. Our evaluation shows that SMinder has an accuracy of 87%, and we demonstrate its superiority over existing solutions by comparing it with two typical existing approaches. We implement a prototype system and deploy SMinder in 6 drivers and 4 passengers for a month. The feedback from the users are very positive.

The rest of this paper is organized as follows. We confirm the motivation in Section 2 by a online survey. Section 3 gives the detailed design and implementations of SMinder. Our evaluation is reported in Section 4. Section 5 discusses the related work. Section 6 concludes the paper.

2 Motivation

The idea for SMinder originated from the authors' own bad personal experience when forget the smartphone in the car. To motivate a tool to remind the user to take the phone when getting off the car, however, the authors needed to answer the three questions we mentioned in Section 1. These questions were meant to establish the desirability of SMinder, before taking steps to implement such a system.

To answer these questions, we carried out a nation-wide online survey in China[4] using mtc.baidu.com. All questions were multiple-choice. In order to filter out less reliable responses (possibly due to respondents not paying enough attention or simply providing random answers), each survey contained 5 randomly-placed repeated questions with reordered choices. We only considered responses that showed consistency across all the repeated questions. We ran our survey until 1000 valid responses were received. Each participant was aid 5RMB for completing the survey, which is consistent with prevailing compensation rates on mtc.baidu.com. The survey engine had mechanisms to prevent repeated entries by the same user or robot entries.

Survey respondents covered 50 cities in 18 different provinces in China, of whom 48% were male and 53%

[3] Accelerometer is used when the smartphone do not have a barometer sensor.

[4] China has the largest smartphone industry in the world since 2009

female, ranging from 18 to 58 years of age (mean 34.2 and tandard deviation 11.7). Most respondents were frequent drivers. Urban white-collar workers appears most. Specifically, 56% say they drove every day, 35% said they call uber, Didi[5] or taxi more than 2 times every week, where as only 8% said they travel in a car less than 2 times a week. Most of the driving was associated with work, shopping and occasional entertainment.

Often forget smartphone in car? The majority of driver respondents said they had left the smartphone in the car more than once every month. In those drivers, 95% said they have left their smartphone in the car more than once in the past 3 month, and 35% said this happens in every week. In those passengers, 98% said they are worried about leave things back (including smartphones) in a taxi, and 65% said they had left something in a taxi, and as much as 10% said left the smartphone in the taxi caused big trouble.

Interest in a reminder to take the smartphone? When asked whether they would find a reminder useful, 95% of the respondents said they would like to have a tool which will remind them to take the phone when getting off the car, of whom, 3/4 said they would choose a tool as a phone application with low power consumption, while 1/4 said they would choose a tool with a wearable beacon and a phone application. The top 2 concerns are the accuracy of the reminder and the energy consumption of the remind tool.

We were especially interested in finding out how the demographics correlated with interest in SMinder. Taking survey responses as ordinal values, we computed the correlations between these responses and interest in SMinder. Statistically significant positive correlations were found between interest in SMinder and each of (i) being a frequent driver or taxi passanger, (ii) Using smartphone often in the car, (iii) female driver or taxi passenger, and (iv) Used to put the smartphone in the car when driving. This means that individuals with more driving and passengering are precisely those who need SMinder more. Not surprisingly, users who often use smartphone in the car need the reminder. We also find that females forget smartphone more often than males. And those often take the phone out are more likely to forget the smarthpone in the car.

How much power consumption is acceptable? People care about the power consumption of every application running on his phone, if an application cost too much energy and affected the user's charge cycle, the user may choose not to use it. In our result, 80% of all respondents said the acceptable power consumption is 3-5% every day. Specially, some frequent drivers can accept a little more power consumption to 10%. There was a statistically significant correlation between accepting higher power consumption and liking SMinder, as well as being a frequent driver or passenger.

We summarize three key observations from the above results. First, the intuition that drivers often forget their smartphone in the car is corroborated by survey data. Second, users need a reminder to alert them when getting off the car, especially the frequent drivers and passengers. Finally, if the power consumption is less than 5% every day, the users are very likely to have this application running in his smartphone. The last observation was important to us because SMinder is a automatic detection tool that will keep on consuming the smartphone's power. Hence, it is a important fact to determine whether users will accept it. The above results complete our motivation for SMinder. Next, we describe system design, implementation, and actual deployment-based evaluation of accuracy and usability.

3 System design

SMinder operates in two modes. The first mode is the low-power mode. SMinder collects barometer data for car moving detection once every five minutes. When SMinder detect that the user is in a moving car, it switchs to the running mode, where SMinder collect sensor readings for context detection. The recognized contexts is provided as input to our reminder decision mechanism which will decide when to remind the user by vibration and alarm of the smartphone. When SMinder detects that the user is not in a moving car, it will switch back to the low-power mode for energy saving.

Our goal is to use the smartphone sensors to detect a left-behind phone and remind the user. The approach is to detect smartphone contexts and design a reminder decision mechanism to decide when to remind the user. This naturally leads to the following two questions:

What and how to detect the contexts? The goal context we need to detect is that the user is getting off the car without the smartphone. It is a complex context and cannot be directly detected by sensor readings from the smartphone. Our approach is by context inferring based on context sensing. Smartphones now have more sensors embedded and can be used for context detecting. For energy saving, we only choose low-power sensors like accelerometer, light, and barometer (0.0488mW at 10HZ e.g., accelerometer). Sensors like GPS and Bluetooth cost too much energy (150mW at 1HZ e.g., GPS). The noisy sensor readings and complex environmental changes increased the detection difficulty. Fortunately, using our

[5]Didi is the China's leading taxi-hailing application.

specially designed algorithms, some basic contexts can be detected, smartphone placement, car moving, car idles/stops, door opens/closes, for example. These contexts show unique patterns in the sensing data. We will show approaches for context detection in the rest of this section.

How to design the reminder decision mechanism? The reminder decision mechanism is a context inferring engine based on base contexts. In our approach we analyze the context change in different scenarios when leave a phone behind. For example, a driver stops the car engine, opens the door and leaves the car. The phone didn't move after the the door opened, and we need to remind the user. In our approach, we observe the contexts of different scenarios need to remind the user and formalize them as inferring rules. The reminder decision mechanism works based on the rules and decides when to remind the user.

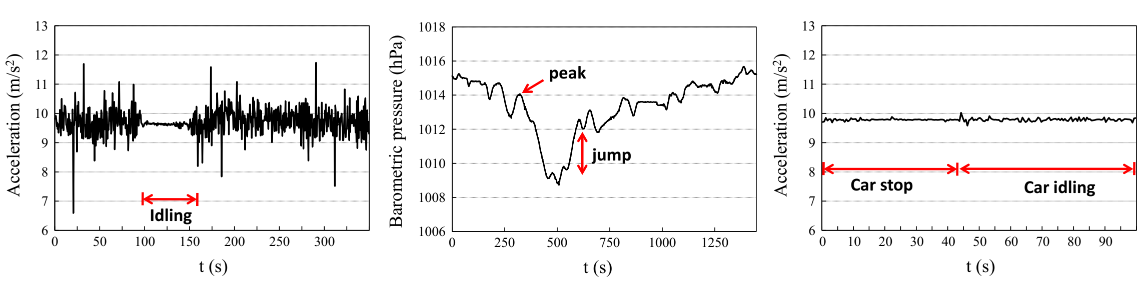
3.1 Context detection

With smartphones now reaching the computational power of personal computers, they are expected to behave intelligently: they should silently understand what the user is doing, help in ongoing or future tasks, and adapt accordingly. A key ingredient of such intelligent smartphone behavior is context-awareness. The phone needs to continuously understand what the user is doing. Context is typically derived from the multitude of sensors on the phone. In this section, we will introduce how to use sensors to detect the contexts. In our approach, the context detection algorithms are designed to be not complex because it should be online and energy efficient. The result shows that they are accuracy enough for our reminder decision mechanism.

3.1.1 Car context detection

In order to detect a user is getting off the car, SMinder needs to understand the basic car contexts, including car moving and car stops/idles. This problem is a part of Transportation mode detection, which is a special case of context-awareness where phone automatically understands the user's daily commute. Being one of the lowest-power sensors on the phone, accelerometer is predominantly used sensor in transportation mode detection [7]. To detect car moving, we use Google's Activity Recognition API [1], which is an application that maintains an activity diary for the user. It is an accelerometer-based context detection algorithm, it is part of the Google Play API, and is capable of detecting user in a moving car. Figure 1a is a typical data trace when a user is in a moving car. The variation is smaller than walking, but larger than standing.

Since the phone's battery life-time is critical, context-detection algorithms must run at extremely low-power. In our approach, at most times, SMinder is in the low-power mode and continuously doing car moving detection every five minutes. Although accelerometer is a low power sensor, to detect car moving activities accurately, sampling rate is typically $10Hz$ and above. The high sampling rate, three axial directions, and position dependence make the classification complicated and increase power consumption. We present an alternative approach to detect car moving using only the barometer. It is inherently orientation and position-independent can be used with a low sampling rate of $2Hz$. We demonstrate that the barometer can be used for car moving detection at extremely low power. The detection is based on the intuitive logic that users in cars see more rapid changes in height, including larger number of ups and downs. Our approach is similar to [11] where we detect jumps and peaks in barometer data. As shown in Fig. 1b, we define a *jump* as a height change of more than 0.8 meters in 5 seconds. For every height reading from barometer (2Hz), we check if there are jumps in height by calculating the difference with a height reading obtained 5 seconds earlier. Peak is based on the observation that cars show a larger number of ups and downs in a given period of time compared to a person on foot. We use a simple online algorithm to detect a peak. In summary, in a 60 seconds time window, if there contains more than 2 jumps or at least 1 peak, we conclude that the user is in a moving car. In SMinder, barometer based approach is used unless the phone does not have a barometer sensor.



(a) Accelerometer readings in a car (b) Barometer readings in a moving car (c) Acceleration when car stops and idles

Fig. 1 Car state detection

When SMinder detects the car moving context, it is easier to detect the car idling/stop state. As shown in Fig. 1c, when the car is idling, the accelerometer data keep stable with a slight variance, for the engine is still working. When the car engine stop, the accelerometer data should stay unchanged. In the figure we find that the readings still have a slight variance, due to the noise of the sensor itself. In our approach, we do not distinguish idling from stop when the user is in the car. If the variance of accelerometer data is smaller than 0.2 m/s^2, we conclude the car is now idling or stopped. Sometimes can't detect a car stops because the user move the phone immediately after he stops the car, this is acceptable in our approach because we only need this when the user forgets to take the phone.

In order to evaluate the accuracy of our online algorithms for car context detection. We collected two days of sensor data from three users who drive to work every. The data collection begins from 7 AM to 8 PM everyday. The driving time is logged for groundtruth, and is about 7 hours. Besides the driving, the data contain accelerometer and barometer readings of user driving, walking, running, going up/down stairs, standing and sitting. The accelerometer based car moving detection accuracy is 87.3%, and the barometer based accuracy is 89.5%. We also evaluated car stops/idles detection during a two hour driving. The result shows that if we assume the car moving detection accuracy is 100%, the car stops/idles accuracy is 93.8%.

3.1.2 Smartphone state detection

The smartphone states are very important for SMinder. Here we detect two states, phone placement in the car and phone movement after car stops or idles.

For phone placement, if we detect that the smartphone is placed in the car, but not in user's pocket or bag, we are more likely to remind the user. Another typical example is the phone slipped out of the pocket and the user didn't realize, this happens often for passengers in a taxi. In our approach, we use the sensor and phone usage information to detect whether the phone is in pocket or placed in the car. At daytime, the light intensity is much lower ($< 1lux$) in pocket than outside the pocket ($1000 - 100000lux$) in the car. Such a phenomenon can still be observed when the light sensor is rotated toward the car seat ($> 100lux$) at daytime. When the light intensity shows that the phone is outside of the pocket and the phone is locked, SMinder concludes that the phone is placed in the car. Moreover, if there is a light intensity change from zero to high, and the phone is locked (user is not using it), the phone is probably slipped out of the pocket. Under these situations, when we detect the car door opens, SMinder will remind the user. At nighttime, the light sensor is not sensitive enough to distinguish whether the phone is in or outside the pocket because the light intensity are all bellow 1 lux. Besides the light sensor information, the usage of the smartphone can also imply the placement of the smartphone. For example, drivers often use the smartphone for navigation, and it should be placed in the car. Another example is when the phone is connected to the power supply. We use the Android API to get the running application, power state and user action of unlocking the phone by registering an event listener.

In summary, we have two rules to detect whether a smartphone is placed in the car. 1) The light sensor reading is larger than $50lux$ and the smartphone is locked. Here, we need to make sure that the smartphone is locked because if the phone is unlocked, the user must be using the smartphone on hand. 2) The phone is running a navigation application or the phone is connected to the power supply.

When the user is getting off the car, it is important to detect whether the phone is moving with the user. If the user is taking the phone, the accelerometer readings will show the signature of user movement, as show in Fig. 2. If the user does not take the phone, the accelerometer reading will keep stable. When the car state is stop or idling, we detect whether the user is getting off the car by the accelerometer variance in the next 5 seconds, if the variance is larger than 1 m/s^2, we conclude that the phone is moving with user; Otherwise, SMinder will remind the user. To evaluate the accuracy, we carried out about 100 test cases of user getting off the car, 50 cases take the phone, and 50 didn't. Assume we already know that the user is getting off the car, the taking phone detection accuracy is 92%. The fail cases are mainly because the user didn't take the phone, but it is moved because of some other reasons.

3.1.3 Car door context detection

In order to detect the user is getting off the car, the action of door opens and closes is useful. We try to use the barometer sensor to detect this context. We observe that barometer

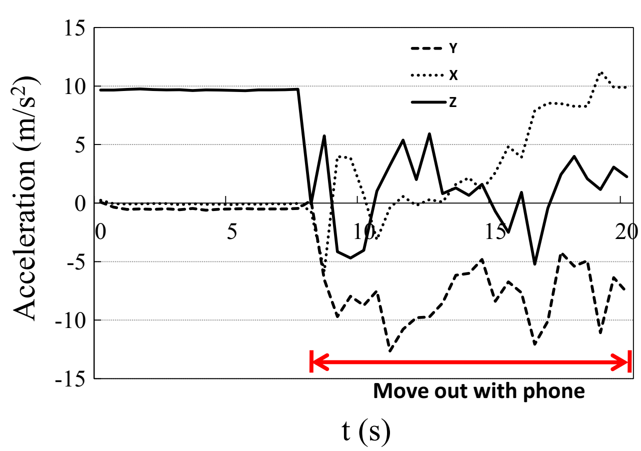


Fig. 2 Accelerometer readings when moving out of the car

Fig. 3 Barometer readings when car door opens and closes

shows a clear signature in the process of door opens and closes. With air conditioning or ventilation equipment used in a car, barometric pressure appears different between the inside and outside of the car. For this reason, when the car stops and opens the door, there exists a sharp drop in barometric pressure. On the contrary, there exists a sharp increase in barometric pressure when the door closes. This change appears clearly in the readings from the barometer sensor. We show this in Fig. 3. When the car stops/idles state is detected, we check whether a door opens or closes event has occurred using barometer readings. In Fig. 3, the readings experience a sudden drop for about 0.2hPa when the door opens, keep stable for some seconds and experience a sudden increase for about 0.2hPa when the door is closed. After the car stops/idles is detected, if there is a barometer drop or increase of more than 0.1hpa, we conclude that the door opens/closes.

The driver sometimes may open the window or close the air conditioning system when driving. In this situation, the door context cannot be detected because there will show no barometric pressure change when opening or closing the door. Here, we need another reliable approach to detect the door opens context. The observation is, when the car stops or idles, the magnetic field around the smartphone keeps stable. If the user opens or closes the door, the magnetic field will be disturbed and changed since the door is made of metal and is near the smartphone. Our experiment data is shown in Fig. 4. The magnetic field readings show clear variance when the user opens and closes the door. The readings change when the door opens, and return to the original value when the door closes. One may worry about the magnetic field reading will be affected by passing by cars, our experiment data show that the influence is small, mainly because the lateral distance between cars in the road is too far to impact the magnetic field reading of the smartphone in the car. In detail, when the car stops/idles is detected, and the accelerometer readings keep stable (the user is not moving it), we monitor the magnetic field variance. If there is a variance of more than $0.1\mu t$, then we conclude that the door opens/closes. In the 180 test cases in our evaluation, if the user opens the door nearest to the smartphone, it can be detected with an accuracy of 87%.

As the input of the context detection algorithms, the sensor readings actually contain some noise. The noise sometimes appears as some isolated points and in most of the time, the noise appears as the jitter and rough of the sensor readings curve. To deal with the noise, we first filter the isolated points based on the average value feature. After that, we smooth the values with a reasonable window based on the type and sampling rate of the sensor. Readers may have a question. What if the user moves the smartphone when the car stopped? If the user moved the smartphone, this will affect the accelerometer and magnetic field readings. The car stops/idles detection and car door opens/closes detection will all fail. SMinder cannot remind the user. There is no need to worry because the user is taking the phone, and we certainly do not have to remind him.

3.2 Reminder decision mechanism

Using smartphone sensor based context detection, we can detect the contexts of the car and smartphone. In SMinder we remind the user using rule based context reasoning on a state machine. As shown in Fig. 5, there are totally 5

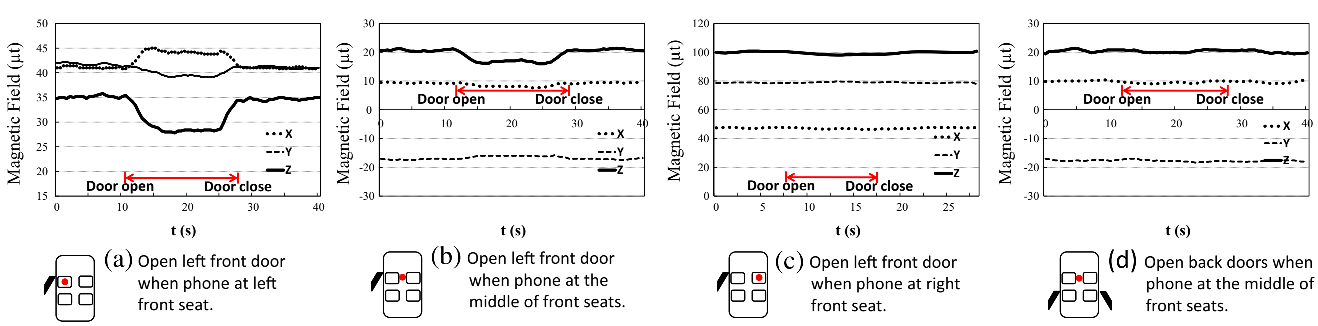


Fig. 4 The magnetic readings when the car's door opens

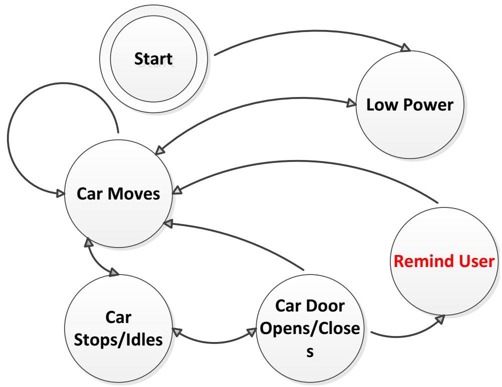


Fig. 5 The state machine of reminder decision mechanism

states, including low power, car moves, car stops/idles, door opens/closes, and remind user. The state changes between car states are trigged by our car and door context detection algorithms, which we have already introduced. The remind state, which means that SMinder decided to remind the user to take the phone, is trigged using our rule based context reasoning using context detection results.

Before showing the basic rules in this approach, we have the following assumptions, 1) In most cases, when the door nearest to the place of the smartphone opens, the phone's owner is getting off the car. This is because most users put the phones near themselves and so near the doors. If the user is getting on the car, but the phone is already in the car not happens often. For example, the user wants to take his left-behind phone. 2) Every time the user gets off the car, he should bring his smartphone. This assumption holds because most users will not leave their smartphone in the car, even with a short time. Under these assumptions, we have the following observations, formalized as context inferring rules.

Observation 1: When the car door nearest to the place of the smartphone opens, the owner is getting off the car, if the smartphone is placed in the car, we should remind the user to take the phone. We formalize this observation as a context inferring rule as follows.

Rule 1 *When smartphone detects that a door opens and the smartphone is placed in the car, SMinder will remind the user immediately.*

Formally, given that

1) C_2: DoorOpen = true;
2) P_1: Smartphone Placement = inCar;
3) A_0: Remind the user;

$R_1 : C_2 \wedge P_1 \rightarrow A_0$.

Observation 2: When the user is getting off the car, if we do not know whether the smartphone is placed in the car, but we find out that the smartphone does not move with the user, the smartphone is very likely to be left behind. This is formalized as follows.

Rule 2 *When smartphone detects that a door opens and does not know whether the smartphone is placed in the car. If the smartphone does not move with the user, SMinder will remind the user.*

Formally, given that

1) C_2: DoorOpen = true;
2) P_2: Smartphone Placement = unknown;
3) P_3: The smartphone does not move in n seconds[6] after C_2;
4) A_0: Remind the user;

$R_2 : C_2 \wedge P_2 \wedge P_3 \rightarrow A_0$.

Observation 3: When the smartphone slips out of the pocket to the car, the user needs a reminder. In this situation, the smartphone's position is changed from a dark place (pocket, bag) to a bright place and the user does not unlock the phone.

Rule 3 *In a moving car, the light intensity goes through a raise from dark to bright and the user does not unlock the phone in the next few seconds, SMinder will remind the user.*

Formally, given that

1) C_1: The car is moving;
2) P_4: The light intensity goes through a raise from $(< 1lux)$ to $(> 100lux)$;
3) P_5: User does not unlock the phone in 5 seconds after P_4;
4) A_0: Remind the user;

$R_3 : C_1 \wedge P_4 \wedge P_5 \rightarrow A_0$.

Based on these rules, SMinder knows when to trigger a reminder for the user. For detail, we use the flow chart in Fig. 6 to show the working process of our reminder decision mechanism.

[6]The default value of n is 5 seconds, and can be optimized based on user habits.

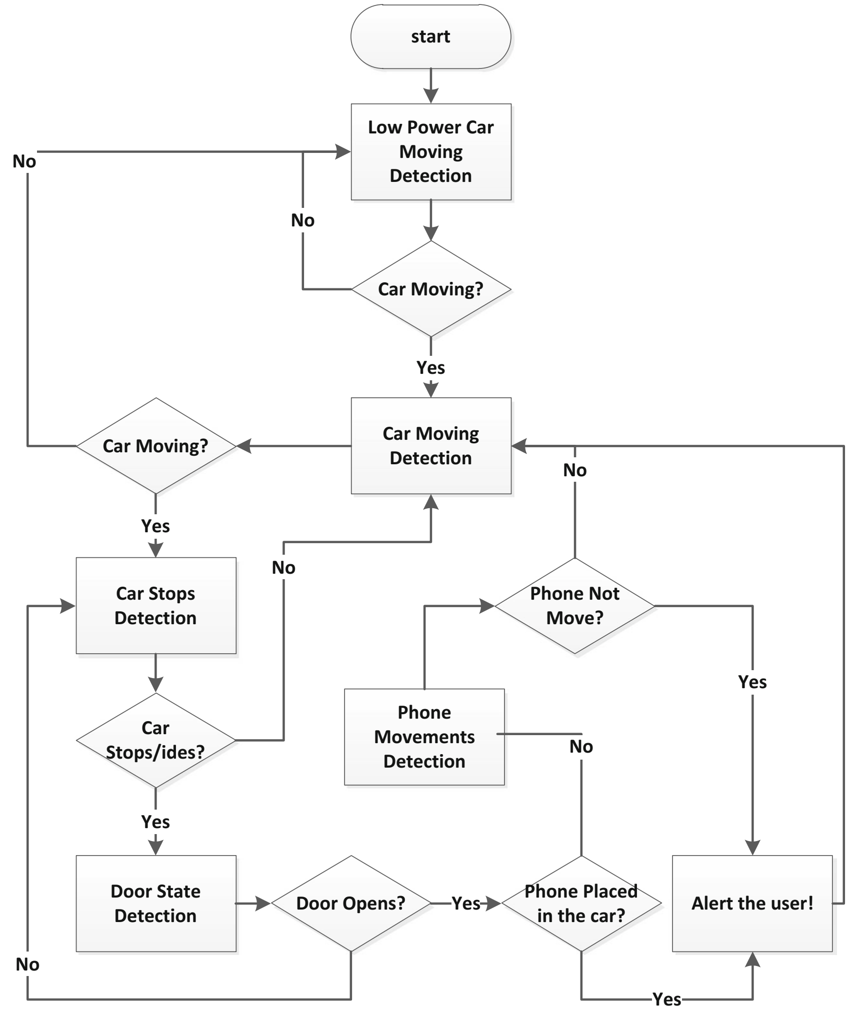


Fig. 6 The flow chart of Reminder Decision Mechanism

4 Evaluation

We evaluate our work based on accuracy, power consumption, and user experience, comparing it with two other approaches.

1) The apple iBeacon based approach iBeacon [9] is a protocol developed by Apple and introduced at the Apple Worldwide Developers Conference in 2013.[7] The iBeacon-compatible hardware transmitters - typically called beacons - a class of Bluetooth low energy (BLE) devices that broadcast their identifier to nearby portable electronic devices. The technology enables smartphones to detect iBeacons in close proximity. In this approach, the user wears a beacon on body, tied to the keychain, for example. The smartphone detects the distance with the beacon periodically based on the received signal strength of the beacon. When the user leaves the car without the smarphone, the distance reach a certain threshold and the smartphone or the beacon will alarm. This approach has been widely used for bidirectional find things. In order to evaluate its performance, we developed a prototype application with the same idea to Chavan's approach [2].

2) The patent US 9070276 B2[8] This is a patent for method and apparatus for detecting a left-behind phone in a vehicle. The main idea is to use a vehicle computing processor, it is configured to, through wireless communication with the smartphone. The processor knows the state of the vehicle engine (running or stopped) and doors (open or closed), and uses the accelerometer readings from the phone to determine whether the phone is moving with the user. If not, the processor will alert the user. For simplicity, we assume that there exist a vehicle computing processor which collect the accelerometer readings from the smartphone. In our implementation, the vehicle computing processor is a laptop, it is connected to the user phone through Bluetooth. The laptop gets the real-time vehicle engine and door states through the OBD port of the car.

[7]iOS: Understanding iBeacon. Apple Inc. February 2015.

Our evaluation contains two parts. The first part is the well-designed experiment to evaluate the accuracy and power consumption of the three approaches. The second part is to provide this application to the volunteers for use in their daily lives, and collect their feedback about the user experience. In the first part, there are 6 participants and 3 cars. They are divided into 3 teams, each team contains one car and two participants (a driver and a passenger). We choose a compact car, a SUV and a MPV to cover different types of cars. The smartphones they used are Galaxy S3, Galaxy Nexus, Galaxy S6 and Nexus 5X. The experiment lasted five days, 4 hours at daytime and 2 hours at nighttime every day. We evaluate the driver and passenger respectively, each for 3 hours a day. For the driver, the process is 1) stop the car engine, 2) open the door, 3) leave the car, and 4) close the door. For the passenger, the process is 1) stop and hold the car, 2) open the door, 3) leave the car, 4) close the door, and 5) the car leave. Every time the user leaves the car, the user may C1): puts the phone in pocket and leaves the car with phone (20% cases), C2) puts the phone in the car and takes the phone when leaving the car (10% cases), C3) puts the phone in the car and does not take the phone when leaving the car (70% cases). During the experiment, when one user does the leave car activity, the other user will record the ground truth using our specially developed application, user only records the user level ground truth such as leave car without phone, remind success, remind fail, .etc. The low level ground truth such as car stops, door opens and closes are recorded by the laptop through ODB connection to the car. After a test case, the user goes back to the car and have another try after driving 3-5 minutes. The phones used by volunteers include Nexus 6, Nexus 5x, Nexus 5, and Galaxy S6, all with a version higher than Android 4.1. In the first part, we evaluate our approach for three days, the iBeacon based and patent based approaches are evaluated in the rest two days, each for one day.

In the second part, there are 10 volunteers, including 6 drivers who drive the car to work at working days, and 4 passengers who go to work by car, but not as the driver. The smartphones fill in to Google, Samsung, HUAWEI, and HTC. We developed the application of SMinder using Android ADT integrated development environment. It is a small APP with a size of only 285KB. The volunteers download the App on the phone and it runs in the background during their daily commute. No special instructions were given to the users. The test lasted for a month. The detection results are logged in the smartphone, and we collect the feedback from all the users.

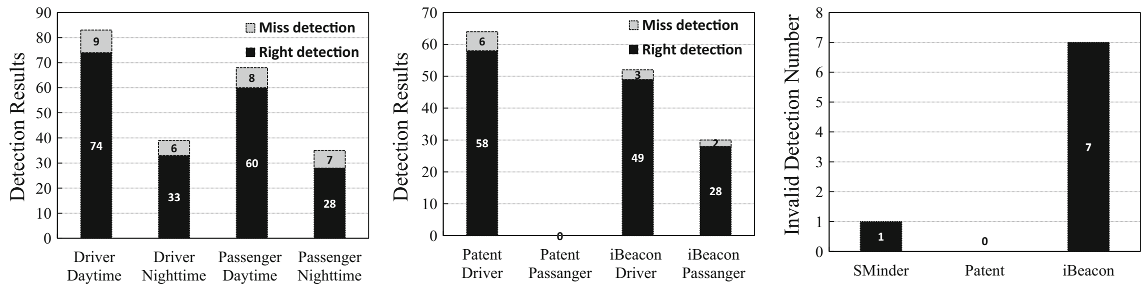
4.1 Accuracy

To evaluate the accuracy of the three approaches fairly. We make use of the following metrics about accuracy: 1) False negative rate (Miss detection rate): it is calculated as the fraction of the total number of left-behind phone cases that the system is not identified. 2) False positive rate: it is calculated as the fraction of the total number of non-left-behind phone cases that the system is identified as left-behind phone cases. 3) Invalid detection performance: shows the frequency of the system gives invalid reminders in the daily life.

Table 1 compares the accuracy metrics of the three approaches. For SMinder, there are totally 225 left-behind phone cases and 195 detected 30 missed, the False negative rate is 13.3% (86.7% accuracy). For detail, we separately show the results for driver and passenger at daytime and nighttime in Fig. 7a. The accuracy at daytime is better than the nighttime. By analyzing the logged data of miss detection cases, we find that mainly because the night intensity based phone placement detection cannot work at nighttime. The result also shows that SMinder work well for both drivers and passengers. The patent based approach got 64 left-behind cases and the False negative rate is 9.3%, better than SMinder, but it can only detect the driver, as show in Fig. 7b. The passenger is not supported because the passenger's smartphone cannot conveniently connects to the driver's car. We had 82 test cases for the iBeacon based approach and the False negative rate is 6%, as show in Fig. 7b. It shows the minimum miss detection rate, but the False positive rate is very high which will significantly affect the user. The iBeacon based approach shows the highest False positive rate 38%, this is mainly because the Bluetooth signal is not stable, even when the user is in a moving car, it may falsely reminder the user because of the fluctuation of the Bluetooth signal. For SMinder, there are totally 85 non-left-behind phone cases, including 8 false positive reminders, the False positive rate is only 9%. For patent based approach, the False positive rate is 5%.

Table 1 Reminder detection accuracy

Approach	False negative rate		False positive rate		Invalid
	Driver	Passenger	Driver	Passenger	performance
SMinder	12.3%	14.5%	8.3%	9.6%	Medium
Apple iBeacon	5.7%	6.7%	37.4%	38.8%	Poor
Patent US9070276	9.3%	−%	5%	−%	Good



(a) Accuracy of SMinder

(b) Accuracy of iBeacon and patent based Approach

(c) Average invalid reminders in a day

Fig. 7 Accuracy of the three approaches

When the user is not in the car, in the three days experiment, as shown in Fig. 7c, SMinder averagely gives two invalid reminders a day, means very limited negative impact to the user, we believe the invalid detection performance is medium as shown in Table 1. The patent based approach gives no invalid reminder because it stops working when the user leaves the car, and the performance is Good. The iBeacon based approach gives averagely 16 invalid reminders a day, which is not acceptable for most of the users, and the performance is poor.

4.2 Power usage

In this section, we evaluate the energy consumption of SMinder, and compares it with two other approaches. Measurements were performed on Nexus 5X using the Monsoon Power Monitor. For all three approaches, the applications are all run in the background after acquiring a wake lock to keep CPU processing on. The screen is turned off and the phone is switched to flight mode. The Android OS is unmodified version 6, which is the original factory edition.

For SMinder, in most of the time, it is running in the low-power mode. It awakes cpu every 5 minutes and collects barometer data 60 seconds for in car detection. Barometer data is sampled with rate 2Hz using the Android sensor manager API. If the user is detected in a running car, SMinder switches to running mode, it keeps cpu on and collects accelerometer and light with the rate of 10HZ and 1HZ. The online context detection algorithms is running. The iBeacon based approach keeps the Bluetooth on, scans and calculates the distance with beacon every 5 seconds. The patent based approach works when the smartphone is collected to the car (the laptop in our implementation) through Bluetooth. When the car stops, the smartphone will sample and send the accelerometer data to the car through Bluetooth. We only compare the power consumption of the smartphones here, we need to know that SMinder only use the smartphone and do not need any infrastructure. Compared to SMinder, the iBeacon based approach requires an iBeacon device on user's body, which needs a button battery to provide power. The patent based approach needs a vehicle computing processor to work, which consumes the power of the car.

Table 2 shows the power consumption. The power values listed for the three approaches include the base CPU awake power, sensing power, as well as computation power. The CPU idle power consumption is removed because we only measure the extra power consumed by the approaches. We measure the power consumption in two scenarios, when the user is in and not in a car. When the user is in the running car, SMinder is in the running mode, the power consumption is 128 mW, which is higher than the other two approaches. When the user is not in the car, SMinder works 1 minute in every 6 minutes, actually, SMinder collects barometer for 60 seconds and runs the algorithm in the last second, the average extra energy consumption is 51 mW. The iBeacon based approach keeps the Bluetooth on and checks the distance in every 5 seconds. It has the same energy consumption when the user is in or out of the car. The patent based approach needs to keep the Bluetooth on to wait for connection, once connected, it needs to collect and

Table 2 The power consumption

	CPU Idle (mW)	CPU Awake (mW)	Low Power Model Not in Car (mW)	Running Model In Car (mW)	18 hour extra Energy Consumption (mWh)	Percent of Battery
SMinder	25	85	51	128	570	4.65%
Apple iBeacon			93	93	1249	10%
Patent US9070276			43	122	428	3.5%

transfer the sensor data at real-time. The remind alarm costs the same energy for all approaches. It is acceptable since it appears not often and the lasted time is very short.

In order to compare the average power consumption, we assume the application runs 18 hours a day, from 6 AM to 12 PM, and the time a user in a car is averagely one hour a day. So, we can get the average power consumption of a day for the three approaches. As shown in Table 2, SMinder costs more power than the patent based approach, but is more energy efficient than the iBeacon based approach. For a typical 3220 mAh and 3.8 V smartphone (Nexus 6), the extra energy consumption of a day is averagely 4.65% (570 of 12236 mWh), this is acceptable for the user.

4.3 User feedback

We now discuss users' actual end-to-end experiences as they use SMinder. We look at the user feedback about remind accuracy, energy consumption, and how they felt about using SMinder. As users completed our study, we conducted a simple exit interview to ask them about their general experience regarding using SMinder, expressed in their own words. Of all 10 volunteers. First and foremost, SMinder remind the drivers accurately, 5 of the 6 drivers said they forget the phone in the car for more than once and SMinder successfully reminded them before they lock and leave the car. In the 4 passangers, SMinder successfully reminded once, although they didn't really forget the phone in the car, the passengers said SMinder accurately reminder them when they open the door and the smartphone is placed in the car. The other accidently discovery from the user feedback is SMinder often reminds the female drivers to take their bag with the smartphone in it. For energy consumption, 6 users said they didn't notice the extra power consumption. The other 6 users said the energy consumption is acceptable and it didn't affect their daily usage of the smartphone. After the survey, 3 users said they will sure to use SMinder because they often forget phones or bags in cars. 5 users said they will use SMinder because it is a good tool and the energy consumption is very low. The rest 2 users said they won't use it because they forget the smartphone in car rarely.

4.4 Analysis of accuracy

As shown in Table 1, the FNR and FPR are not perfect. We find out it falls into the following two reasons. The first is the complex environmental change. The situation can never be always the same as we modeled. Since we implement the reminder decision mechanism based on some context inferring rules, there exist some situations cannot be handled by the inferring rules. The second reason is the context detection algorithms. The sensor data are noisy, sometimes the noise is not big and can be handled by our context detection algorithms. Sometimes the noise is too big to be handled and cause false detections. The false detection of the context will cause false input to the reminder decision algorithm which will result in FNR and FPR results. To deal with these problems. The first idea is to deal with the complex working environment of SMinder. We need to optimize our model to contain more situations, including situations that not happening often. Under the new model, there will have more inferring rules and each rule are more dedicated to the situations. For example, the current edition of SMinder does not support the situation when the user is traveling in the subway or in a train. We need to improve the model to support the reminder in a train or subway. The second idea is to improve the context detection algorithms to be more robust when the sensor data are noisy. The way is to use better technologies such as matching learning and deep learning instead of the heuristic approach used in our current edition of SMinder. For instance, the conditions may be a little different based on when and where the car is, and there need some strategies to handle it. Meanwhile, we should also care about the extra cost by the improvement. The complex rules and detection algorithms will increase the computation complexity and cost more energy. The trade off should be carefully balanced because the energy consumption is an important aspect about the usability of SMinder.

5 Related work

In this section, we show the related work in two dimensions. The first is the problem dimension, which is the smartphone anti-lost approaches. The second is the solution dimension, which is context detection algorithms.

Researcher's have promoted ways to prevent users from leaving smartphones in cars. From the point of view of the problem, the problem is a special case of the smartphone anti-lost problem. To avoid lost smartphone at any place, the simplest way is to hang the smartphone on body like an ID, but today few user will choose this way. To solve this problem, some approaches are promoted. The most used solutions are Bluetooth [2, 9] or RFID based approaches. A typical example is the iBeacon approach [9] evaluated in this paper, and the limitations are well discussed. The RFID based approaches [6, 14] are based on the same principle, the user takes a tag on body and the smartphone read the tag to make sure the phone is in close distance. Still suffer the same problem as Bluetooth based approaches. The technology based on car and smartphone communication is also effective. The patent [8] is a typical example. The main idea is to use a vehicle computing processor, it is configured to, through wireless communication with the smartphone. The processor knows the state of the vehicle engine and doors, and use the accelerometer readings from the smartphone to determine whether the phone is moving with the user. If not, the processor will remind the user. This may be the best way to solve this problem. However there are still some obstructions, 1) the approach needs the user connect the car through wireless communication, this is reasonable if the user is a driver, but not suitable for a passenger, a passenger in a taxi for example. 2) It has the facility requirement for the car, where most cars cannot support.

The recent advance of sensors embedded in smartphones has motivated a novel sensor-assisted approach. In this paper, we use the context-detection techniques to detect the left-behind phone context. Context-detection using sensors has been popular for several years, differing in the type of user activities detected, sensors used, and classification techniques. An extensive survey is presented in [7]. Accelerometer is the predominant sensor used. Other commonly used sensors include barometer, light. In the following, we focus on prior work using these sensors for context detection, describing their limitations.

1) Accelerometer: The most popular sensor used for context detection is the accelerometer. Existed approaches extract features from the accelerometer readings and use supervised machine learning to detect user's car activity. Most of the prior work perform the detection offline [5] instead of online, which cannot meet the requirement here. For example, Reddy et al. [10] and [12] implement their classifier on smartphones and perform the training offline. The accelerometer based detection also have the orientation problems, [16] use orientation-independent features to avoid the problem. Accelerometer is also capable of fine-grained classification, it can distinguish different types of vehicle. Authors of [5] can detect user traveling on bus, train, car and subway with low power consumption.
2) Barometer: The increasing availability of barometer embedded in smartphones (e.g., Nexus 4) has motivated researchers a new way for context detection. Although it is first introduced in the android phone for aiding GPS [20], researchers find other applications, floor localization, for example. Due to the barometer's good relative accuracy, it is well used for floor-change detection [15]. Our previous work [19], use the barometer to detect the door opens and closes of the subway train. Similar to our approach, [17] detect door opens and closes in buildings using barometer, and shows good accuracy. To detect a user is in a moving car, our approach is based on [11], who first used barometer for detection of the mode vehicle.
3) Light: The light sensor on smartphone is used to adaptively adjust the screen brightness for energy saving. It detects the ambient light with high sensitive and can be used for context detection. Authors in [3, 13, 18] use the light sensor for indoor position and navigation. In our paper, the light sensor is used for in/out pocket detection.

6 Conclusion and future work

In this paper, we present SMinder, a novel, effective, low power approach to remind user taking the phone when he is getting off the car. SMinder requires neither any infrastructure nor any human intervention. It uses low power smartphone sensors only. SMinder reminds the user with high accuracy and minimum energy consumption, making it more realistic for real-world use. Compared to the existing approaches, SMinder is cheaper and easier to use. For our future work, we will further improve the reminder decision mechanism based on machine learning algorithms. We also plan to provide SMinder as a free service for public use.

Acknowledgments This work was supported by the National Natural Science Foundation of China under Grant No.61702261, the China Postdoctoral Science Foundation under Grant No.2017M621742, and the Foundation of State Key Laboratory of Novel Software Technology under Grant No.KFKT2017B15.

References

1. Google's activity recognition api. http://developer.android.com/google/play-services/location.html
2. Chavan PS (2014) Design and implementation of anti lost bluetooth low energy mobile device for mobile phone . Int J Engineer Res Appl 4(5):73–76
3. Cheok AD, Yue L (2011) A novel light-sensor-based information transmission system for indoor positioning and navigation. IEEE Trans Instrum Meas 60(1):290–299
4. Enck W, Gilbert P, Han S, Tendulkar V, Chun BG, Cox LP, Jung J, Mcdaniel P, Sheth AN (2014) Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM Trans Comput Syst 32(2):393–407
5. Hemminki S, Nurmi P, Tarkoma S (2013) Accelerometer-based transportation mode detection on smartphones. In: ACM Conference on Embedded Networked Sensor Systems, pp 1–14
6. Hussain MJ, Li L, Gao S (2017) An rfid based smartphone proximity absence alert system. IEEE Trans. Mob. Comput. PP(99):1–1
7. Incel OD, Kose M, Ersoy C (2013) A review and taxonomy of activity recognition on mobile phones. Bionanoscience 3(2):145–171
8. Le J, Rao MK, Prakah-Asante KO (2015) Method and apparatus for detecting a left-behind phone
9. Newman N (2014) Apple ibeacon technology briefing. J Direct Data and Digital Marketing Practice 15(3):222–225
10. Reddy S, Mun M, Burke J, Estrin D, Hansen M, Srivastava M (2010) Using mobile phones to determine transportation modes. Acm Trans Sensor Netw 6(2):662–701
11. Sankaran K, Zhu M, Guo XF, Ananda AL, Chan MC, Peh LS (2014) Using mobile phone barometer for low-

power transportation context detection. In: ACM Conference on Embedded Network Sensor Systems, pp 191–205
12. Siirtola P, Rning J (2012) Recognizing human activities user-independently on smartphones based on accelerometer data. Int J Interactive Multi Artif Intell 1(5):38–45
13. Silberman N, Fergus R (2011) Indoor scene segmentation using a structured light sensor. In: International Conference on Computer Vision - Workshop on 3d Representation and Recognition, pp 601–608
14. Tanbo M, Nojiri R, Kawakita Y, Ichikawa H (2015) Active rfid attached object clustering method based on rssi series for finding lost objects. In: Internet of Things, pp 363–368
15. Vanini S, Giordano S (2013) Adaptive context-agnostic floor transition detection on smart mobile devices. In: IEEE International Conference on Pervasive Computing and Communications Workshops, pp 2–7
16. Wang S, Chen C, Ma J (2010) Accelerometer based transportation mode recognition on mobile phones. In: Asia-Pacific Conference on Wearable Computing Systems, pp 44–46
17. Wu M, Pathak PH, Mohapatra P (2015) Monitoring building door events using barometer sensor in smartphones. In: UbiComp, pp 319–323
18. Xu Q, Zheng R, Hranilovic S (2015) Idyll: indoor localization using inertial and light sensors on smartphones. In: ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp 307–318
19. Ye H, Gu T, Tao X, Lu J (2014) Sbc: scalable smartphone barometer calibration through crowdsourcing. In: International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, pp 60–69
20. Zhang J, Edwan E, Zhou J, Chai W (2012) Performance investigation of barometer aided gps/mems-imu integration. In: Position Location and Navigation Symposium, pp 598–604