



The 13th International Conference on Mobile Systems and Pervasive Computing  
(MobiSPC 2016)

## Hybrid battery-friendly mobile solution for extracting users' visited places

Mehdi Boukhechba, Abdenour Bouzouane, Bruno Boucharda<sup>a</sup>, Charles Gouin-Vallerand<sup>b</sup>, Sylvain Giroux<sup>c\*</sup>

<sup>a</sup>University of Quebec at Chicoutimi, Chicoutimi, G7H 2B1, Canada

<sup>b</sup>LICEF research center, Tele-universite of Quebec, Quebec, G1K 9H6, Canada

<sup>c</sup>DOMUS Laboratory, University of Sherbrooke, Sherbrooke, J1K 2R1, Canada

---

### Abstract

Extracting and analyzing outdoor humans' activities represent a strong support for several applications fields, ranging from traffic management to marketing and social studies. Mobile users take their devices with them everywhere which leads to an increasing availability of persons' traces used to recognize their activities. However, mobile environment is distinguished from one to another by its resources limitations. In this paper, we present a novel hybrid approach that combines activity recognition and prediction algorithms in order to online recognize users' outdoor activities without draining the mobile resources. Our approach minimizes activity computations by wisely reducing the search frequency of activities, we demonstrate that our proposal is capable of reducing the battery consumption up to 60% while maintaining the same accuracy as its similar.

© 2016 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the Conference Program Chairs.

*Keywords:* Activity recognition; Battery-awareness; Activity prediction; Trajectory analysis; Online clustering;

---

### 1. Introduction

Mobile wearable tracking devices, e.g., phones and navigation systems, sense the movement of persons represented by positioning records that capture geo-location, time, and a number of other attributes. Sensing is based on a collection of information related to the achieved activity from raw sensor data (GPS, Wi-Fi, RFID, Bluetooth signals,

---

\* Corresponding author. Tel.: +1 581 882 8640

E-mail address: [m\\_boukhechba@esi.dz](mailto:m_boukhechba@esi.dz)

microphone, camera, accelerometers, magnetometers, etc.), these data are used to extract a pertinent information about the current activity and users' visited places [10]. As such, the mobile phone is no longer only a communication device, but also a powerful environmental sensing unit that can monitor a user's ambient context, both unobtrusively and in real time. This context awareness property makes this field a major piece that provides services to a range of application fields such as real-time traffic monitoring [8], social networking and cognitive assistance. However, the limited battery capacity of mobile devices represents a big hurdle for context detection. The embedded sensors in the mobile devices are major sources of power consumption. Hence, excessive power consumption may become a major obstacle to broader acceptance context-aware mobile applications, no matter how useful the service may be.

Activity recognition services that extract users visited places are supposed to operate 24 hours a day, 7 days a week. Searching for users visited places on every moment, like done in the majority of related works, leads to an excessive power consumption that drains mobile's battery rapidly. We will propose in this paper a novel approach that minimizes power consumption by reducing the calculation frequency of activities. The proposed algorithm learns users' habits and chooses an appropriate time to search for their performed activities. For instance, suppose that the user has the habit of going from home to work every morning, theoretically, there is no need to process the user movements on every time he goes from home to work since it represents useless calculations.

In this paper, we will demonstrate an innovative battery-friendly method that recognizes accurately users' activities without draining the battery of their phones, a method that succeed in detecting incrementally users' visited places without any previously fixed threshold, we will also prove that our proposal reduces outstandingly the battery consumption when keeping a same accuracy rate as its similar.

The following sections detail our contribution: Section 2 reviews related works; Section 3 presents our approach in terms of three major components, i.e. activity recognition, prediction and verification; Section 4 describes the experimentation by highlighting two dimensions: accuracy and power saving. Finally, conclusion as well as the expected contributions, are summarized in Section 5.

## 2. Related works

Research community's efforts are increasing day by day to carry out efficient mobile activity recognition systems. CityVoyager presented in [4] is a recommendation system designed for mobile devices, which recommends shops to users based on data analyzed from their past location history. Authors track the visited shops by the loss of the GPS signal, though, it is known that GPS signals frequently become lost in urban areas due to high buildings or due to some special weather conditions, these situations increase the possibilities of false detections. Furthermore, authors claim to propose an approach designed for mobile phones, however, there is no adaptation noted to support this demanding environment, for instance, finding frequented shops requires a heavy manipulation of the historical records of users' visited shops, authors seem to neglect the limited mobile's resources since there is no support for the limited battery life and there is no effort perceived to online detect and find the frequent shops.

In [3], an algorithm is proposed to associate each stop in a user's trajectory to a list of possible visited places and each of these places is associated to a probability, then, depending on the kinds of activities associated with the identified place, the trajectory is classified into a probable trajectory behaviour. This work uses numerous thresholds that are set manually like the minimum duration of an activity, nevertheless, since these parameters may depend on user profiles, this work may be ineffective on large datasets that contain several profiles.

While developing a rich body of work for managing moving objects, the research community has shown little interest to the limited resources of smartphones, for instance, nearly all approaches repeat the same activity research process for every daily activity which leads to an excessive consumption of phones' batteries.

Moreover, nearly all outdoor activity recognition approaches use a fixed activity's minimum duration threshold that represents the minimum time that the user has to spend in the POI (place of interest) to be declared as visited place. This threshold prevents false activity detection like traffic jams. However, previously fixing this threshold will increase error probability, because when set to a small value, it will increase the number of false activities like passing by a POI, and setting it to a high value, will miss detect some short-dwell activities like buying cigarettes at the convenience store. Consequently, we will be (to the best of our knowledge) the first to propose not only a dynamic approach that learns the activity's minimum duration threshold automatically, but to propose a specific threshold for each POI too. Our approach will assign to each POI a minimum duration threshold to be able to detect both the short

and the long activities. Despite that in real world a POI can contain several activities, related works have linked a POI to only one activity, like assigning a mall to shopping, even if it can contain a multitude of activities like going to a restaurant and cinema. Thus, our approach will handle the plurality of activities inside the POIs. We propose in the following, a novel approach to reduce the battery consumption while online recognizing users' visited places, our algorithm will be totally unsupervised without any beforehand fixed threshold.

### 3. Overview of the approach

We assume the person is traceable via a smartphone, the type of users' traces is not important to us since the proposed approach works for any movements type (GPS, WIFI positioning, WIFI, Bluetooth or GSM triangulation, pedestrian dead-reckoning, etc.). We will analyze incrementally a user's motility to extract his performed activities. The main idea of our approach is to minimize the calculation during the analysis process by using a mixture of activity recognition and prediction algorithms.

Our system is divided into three parts; the first part aims to recognize users' activities when they visit places for the first time, the second part is activity prediction where we predict the next activity to avoid processing the activities already recognized, and finally, activity verification which is a post-processing step that aims to verify if the predicted activity is the right one. Suppose that the user has gone from home to work (see Figure 1-A), for the first time when the user visits these locations, we will recognize the two POI linked to the activities staying at home and working in the office by using our activity recognition model presented later.



Figure 1: The three parts of our hybrid approach. (1) Home, (2) Work,

The next time that the user will go from home to work, our approach will not use the activity recognition model to recognize the activity since it represents a significant source of power consumption, nevertheless, it will use only the association rules driven from the prediction model to estimate the next destination, meanwhile, all the GPS points between home and work will be stored without any processing until we confirm that the predicted activity is that one performed by the user by using the verification model (see Figure 1-B).

If we confirm that the predicted activity is that one performed by the user, we delete the recorded GPS points between home and work because the prediction was made successfully, otherwise, we apply the activity recognition model for the whole recorded points to figure out where the user has gone from home. We are going to detail in the following, the three parts of our system.

#### 3.1. Activity recognition

This step aims to explore a user's activities for the first time, like discovering where he lives and works. We propose a novel approach that recognizes not only stationary activities but moving ones too. In fact, usually, person's activities are divided into two behaviours: stationary and non-stationary, where the second one is also divided into two categories moving to reach a goal and moving to do a goal (see Figure 2).

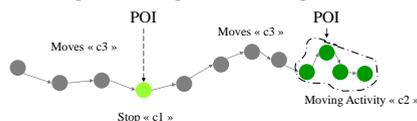


Figure 2. Relation between moves, stops and activities with moving.

For example, working in the office is a stationary activity, while going from work to shopping is non-stationary activity to reach a shopping center, shopping itself is a non-stationary activity too but the goal is to do shopping, so it's an activity with moving. Based on these concepts, we introduce 3 types of clusters that we will recognize

incrementally using our online clustering algorithm:

1. Stop concept, represented by “c1” and characterizes stationary activities.
2. Activity with moving “c2” is a non-stationary activity that requires movement over a time interval.
3. Moves, represented by “c3” are a set of actions that aim to move from a POI to another.

To deal with all these concepts, we present in Fig. 3 the overall approach of our recognition mechanism.

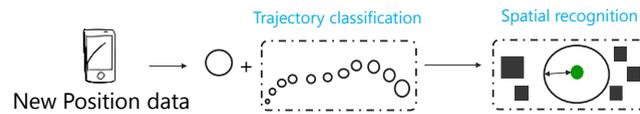


Fig. 3. Activity recognition approach.

We use in the first step an online classification method based on K-means to classify every new GPS data according to the three families (stops, moves, and activity with moving) by using two variables; the user speed and bearing. In parallel, we observe the accumulation of types of clusters, such that, after a certain threshold calculated automatically of the same cluster’s accumulation, we conclude that the person is probably doing something interesting. For the second step, we summarize the accumulated clusters to one probable POI and we start a geospatial research for the closest and the most meaningful geographical entity. If the research process succeeds, we declare this point as a POI (see Fig. 3). For further details about the operating of this approach, please refer to the paper presented in [6].

### 3.2. Activity prediction

The activity prediction step begins by constructing a sequence of *POI* that represents the tracking of users’ daily habits learned using the activity recognition model, every sequence is stored incrementally in a tree structure called Habits’ Tree ‘HT’. On every sequence arrival, our algorithm updates incrementally HT and predicts the next *POI* using the association rules drawn from HT. Every sequence contains a set of disjoint singletons  $POI_j$  and terminates with the end of the day (daily habits). For example, assuming that the user achieved the following activities during a day: home, work, restaurant, work, gym, home; the algorithm will construct incrementally two sequences from these habits:  $S_1$ : Home, work, restaurant and  $S_2$ : Restaurant, work, gym, home. Afterwards, the sequences are stored in HT (see Fig. 5) and the association rules are mined using FP-Tree algorithm [9].

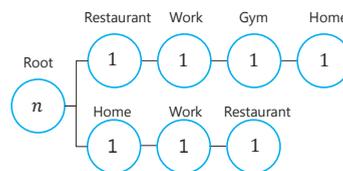


Fig. 4. HT Structure construction. Every node contains a number of occurrence  $n$

After mining the association rules from HT, suppose that we got these rules:  $(work, gym \rightarrow home)$ ,  $(home \rightarrow work)$ ,  $(Home, work \rightarrow restaurant)$ . Predicting the next activity lies on choosing the most appropriate association rules with the greater weight that represent a user’s situation, and using the resulting clause as predicted next activity. For example, if we know that the user has gone from home to work, using the last association rules we can predict that he will go next to the restaurant (see Fig. 4). For further details about the operating of the prediction approach, please refer to the paper presented in [5].

### 3.3. Activity verification

After predicting the next activity, we need to confirm that the predicted activity is that one performed by the user, for this purpose we introduce a new structure of POIs. A POI will become not only a geographic place where the user

carry out an activity, but a geographic entity that is characterized by a minimum duration  $d_{min}$  that represents the minimum duration of an activity, and a distance  $r$  that represents a ray where the activity can be performed. Unlike the related works, we will learn these parameters by adjusting them incrementally and dynamically in function of a user's behaviours (see Fig. 5).



Fig. 5. The  $d_{min}$  and  $r$  characteristics of a POI

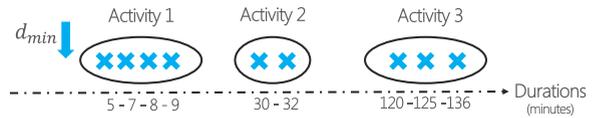


Fig. 6. Activities' duration clustering inside a POI, for instance, activity 1 can be going to convenience store, activity 2 going to fast food, activity 3 shopping.

Our approach handles the POIs as geographic areas that may contain several activities. Each activity is characterized by temporal edges learned from the user behaviours. In this work, we differentiate between activities in the same POI by using only duration, this process is based on a taxonomy of activities durations, for instance, suppose that the POI is a mall that may contain a multitude of activities, watching a movie in the cinema is an activity that its duration is between 1h 30 and 3h, while the duration of eating in a fast food is between 30 min and 1 hour, comparing the users duration of stay with such taxonomy may reveal information on the executed activity, we are not going deeper in the explanation of this step for lack of space.

### 3.3.1. Calculating $d_{min}$ and $r$

Each duration represents a time spent by the user inside or at the surroundings of a given POI, it is calculated using the time of check-in and check-out (see Fig. 5).

In order to calculate the minimum duration threshold  $d_{min}$ , we need to understand how the user behaves inside this POI. As said previously, a user may perform more than one activity at the same place, the  $d_{min}$  calculation starts by regrouping the durations by using Fuzzy C-Means (FCM) [7] because it allows a time duration to belong to more than one cluster which solves the problem of values on the borderline( see Fig. 6).

However, FCM requires a fixed number of clusters, in our case we don't have a prior information about the number of activities in this POI, so, we propose a criterion to calculate incrementally the optimal number of clusters  $C_N$  (see equation 1); the average deviation of each value from the median  $M$  of its most probable cluster. When more than one cluster is analyzed, the criterion value is the sum of each cluster average. The algorithm considers that the optimal clusters number is  $N$  if the  $N+1$  clusters' criterion value doesn't improve significantly the one with  $N$  clusters.

$$C_N = \sum_{i=1}^n \frac{\sum_{j=1}^{n_i} |x_{ij} - M_i|}{n_i} \quad (1)$$

Where  $N$  is the number of clusters,  $x_{ij}$  is the duration of the activity  $j$  in the cluster  $i$ ,  $n_i$  is the number of activities in the cluster  $i$  and  $M_i$  is the median of the cluster  $i$ .

Let the durations in figure 6 be an example to illustrate this clustering step, suppose that we have initially one cluster [5,7,8,9,30,32,120,125,136], its median will be 30 and  $C_1 = 42,6$ . For two clusters, FCM will construct two clusters [5,7,8,9,30,32] and [120,125,136], thus,  $C_2 = 8,5 + 5,3 = 13,8$ , we note that  $C_2 < C_1$ , so we increase the number of clusters to two. For three clusters, FCM will construct the following clusters: [5,7,8,9], [30,32] and [120,125,136], thus,  $C_3 = 1,25 + 1 + 5,3 = 7,55$ , we note that choosing three clusters have improved the criterion  $C$  as  $C_3 < C_2$ . In order to know if we stop at three clusters we have to test the criterion of four clusters, thus, for  $N=4$ , FCM will construct [5], [7,8,9], [30,32] and [120,125,136], consequently,  $C_4 = 0 + 0,7 + 0,7 + 5,3 = 6,7$ .

We note that the gain in the criterion  $C$  is two small for  $N=4$  ( $C_4 - C_3 = 0,8$ , note that we avoid detailing how we judge that a gain in  $C$  is significant or not to simplify our proposal), so, there is no need to add another cluster, and the number of declared clusters is  $N=3$  (see Figure 6).

After clustering the durations inside the POI, it's time to calculate the value of  $d_{min}$ . Remember that our solution is online, which means that  $d_{min}$  can change at every new visit to this POI, so, initially  $d_{min}$  takes the value of the smallest value of the first cluster (for instance,  $d_{min}$  in the figure 6 is 5 minutes), but, at the arrival of a new duration, we compare it to  $d_{min}$ , if it is higher than  $d_{min}$  we declare that the user has visited this POI and we recalculate the criterion  $C$  to figure out if we have to add a new cluster or not.

However, if the duration is less than  $d_{min}$ , the new duration can be a new  $d_{min}$  (the user visited the POI with shorter dwell time) or an error (the user just passed by the POI without performing an activity there), accordingly, we calculate the criterion  $C$  for the new clusters including the new duration, if  $C_{new}$  is significantly higher than  $C_{old}$  we keep  $d_{min}$  and we conclude that the new duration is an error, otherwise,  $d_{min}$  takes the value of the new duration. For instance, let take the previous example presented in Figure 6. If the new duration is 1 minute, the new  $C_3$  will be 8,5, we note that new  $C_3$  is higher than the old  $C_3 = 7,55$ , consequently, we assume that the user has not visited the POI. However, if the new duration is 4,5, the new  $C_3$  will be 7,8 which is not significantly higher than the old  $C_3 = 7,55$ , consequently, we assume that the user has visited the POI, we accept this duration as a borderline duration and we put  $d_{min} = 4,5$ .

The value  $r$  represents a ray where the user has to spend a minimum duration  $d_{min}$  to declare that the user has visited the POI,  $r$  is calculated beforehand in our spatial database for each POI. It covers the total area of the POI including its annexes like parking (see Figure 5).

### 3.3.2. Testing

Remember that the verification process is designed to figure out if our prediction was right and correct it when needed by re-running the activity recognition model (see Algorithm 1).. The prediction error can fall under two cases: the user didn't go to the predicted place and the user has gone to the predicted place but he performed other activities meanwhile. To detect these errors we introduce two types of tests: trajectory duration test and activity duration test.

The first test is to compare the duration of the user trajectory and the maximum of trajectories' durations between the two POIs (the source POI and the predicted POI) to test if the user has gone to the estimated location, if the user's trajectory duration exceeds the maximum of durations we can say that the user has probably gone somewhere else (because he spent more time than usual to reach the predicted POI), consequently, we reapply the activity recognition model because our prediction was probably wrong (see Algorithm 2).

The second test compares the duration of the activity, if the duration of the user's staying in the POI's perimeter (defined by  $r$ ) is less than  $d_{min}$  we can conclude that the user just passed by the area (see Algorithm 2).

---

#### Algorithm 1: the overall algorithm of our approach

---

Input: A user position;  
Output: The Visited POI;  
1: **If** (HT contains current POI)  
2:   Next POI = Prediction POI;  
3:   **Activity verification** (next POI )  
4: **Else**  
5:   Activity recognition until next POI;  
6: **End**  
7:   Update HT;  
8:   Cluster durations inside POI;  
9:   Calculate  $d_{min}$ ;  
10:   Delete GPS records;

---



---

#### Algorithm 2: Activity verification (POI)

---

Input: A POI;  
1: **If** ( trajectory duration < trajectory Max duration )  
2:   Wait until GPS inside POI;  
3:   **If** duration in  $r$  of POI <  $d_{min}$  of POI  
4:   Calculate the new criterion  $C$ ;  
5:   **If** (new  $C \gg$  old  $C$ )  
6:   Activity recognition from the previous POI;  
7:   **Else**  $d_{min} =$  duration in  $r$  **End**  
8:   **End**  
9: **Else**  
10:   Activity recognition from the previous POI;  
11: **End**

---

## 4. Experimental evaluation

We will test our approach's ability to save battery life by comparing our solution to LifeMap application described in [2]. Researchers in LifeMap project collected real traces from 68 persons over four weeks using HTC Hero, HTC Desire, and Samsung Galaxy S smartphones. The tracking application (called LifeMap) was running as a background service to automatically collect the user's mobility and to trace sensor usage time. To collect the ground truth, the participants explicitly labeled the place names and kept a diary of places they had visited with the entrance and departure times. Moreover, the advantage of using such dataset is the ability to compare the power consumption of our method to the authors 'one, since authors tracked the battery status during all the experimentation process.

In this step we will compare our approach to the LifeMap application used to recognize users' motilities, the project can be found in [8], the LifeMap dataset in [1] and the LifeMap mobile application can be found on android play store. We used the LifeMap dataset to test our battery-friendly approach, to do so, we developed an android application that

is fed from LifeMap dataset, the main idea is to make it out as if the users of lifeMap dataset have moved holding our application in their phones, the application recuperates the position coordinates one by one and processes each point using our online approach (activity recognition, prediction and verification), after that we compare the battery consumption of our application with LifeMap application’s one (see Figure 7). In order to bring an objective comparison, we have reproduced the same environment as lifeMap application in terms of the smartphones model and the type of sensors used in the experimentations.

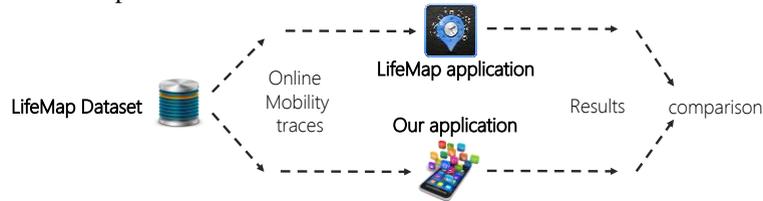


Fig. 7. Test scenario to compare our solution with LifeMap application using LifeMap Dataset.

The total number of recorded hours of battery status in LifeMap dataset is 48900 hour noted from the motilities of 68 persons, however, some of these motilities don’t reflect a user’s motilities in real word, since some users in lifeMap experiment did not have a repetitive behavior during the experiment (several visits of the same POI), consequently, we have chosen 5 users that had the most regular motilities to reflect fairly the real word situation.

The total number of hours experimented from these 5 users is near 2900 hour, we have used five smartphones to record the power consumption of our hybrid approach for each user and compare it to LifeMap results. Due to insufficient space, we present in figure 8 the tracking of one user’s battery life for 72 hours using LifeMap and our hybrid solution, however, the results derived from the total 2900 hours will be presented in Table 1 and figure 9.

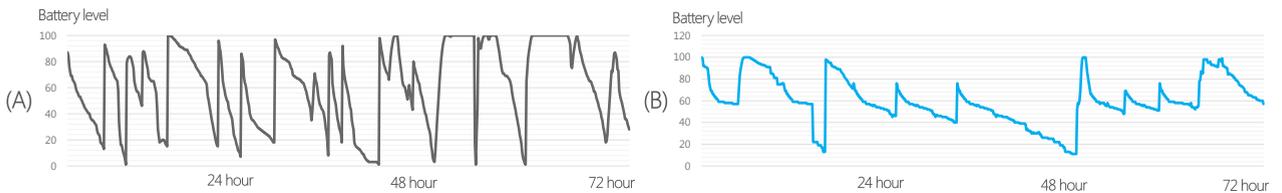


Fig. 8. Results comparison between LifeMap (A) and our hybrid solution (B) during 72 hours of activity recognition.

Our approach shows an interesting battery saving capacity, we notice from figure 8 that our approach needed only 3 to 4 battery recharges contrariwise LifeMap that needed more than 10 recharges during 72 hours. However, the number of recharges is not an efficient indicator that quantifies the power consumption, since, like noticed in figure 7, users tend to recharge partially their phones, therefore, we have introduced a new indicator to quantify the power consumption called *PC*. We put  $PC = T_r/T_d$ , where  $T_r$  is the global battery recharge time and  $T_d$  is the global battery discharge time, note that we exclude the time where the battery was full but still under recharge because it can falsify the results presented in table 1.

Table 1: *PC* and accuracy comparisons between our hybrid solution, our activity recognition model and LifeMap application

	Our hybrid solution	Our activity recognition model	LifeMap
<i>PC</i>	9,4%	15,9 %	16,7 %
Accuracy	76,5%	77 %	73%

The *PC* comparison between our hybrid approach and LifeMap presented in table 1 confirmed that our solution saves the battery’s life by about 45% while keeping a better accuracy than LifeMap’s one (73%) (Accuracy= Correct/tested activities) and the same accuracy as if we applied the activity recognition model continuously (77%), this is justified by the fact that our hybrid approach has the ability to recognize the errors generated by the predictions, and to correct them by reapplying the activity recognition model, consequently, the hybrid approach acts like if we have applied the activity recognition model all the time, but with much less power consumption. We tracked in figure 9, the average daily *PC* value of the 5 users during 23 days, to go deeper in the analysis of the *PC* indicator.

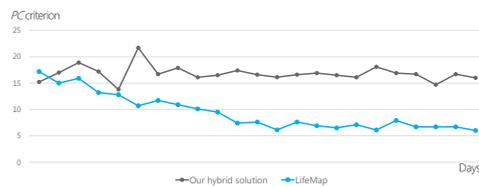


Fig. 9. The average daily PC value of the 5 users, comparison between our solution and LifeMap for 23 days.

We notice that the *PC* value of LifeMap is stable contrariwise our solution that starts from a value of 17% to fall under the 7%, this is justified by the fact that our solution consumes more power in the first times of the experimentation because it applies the activity recognition model frequently in a perspective of learning the user's habits, when done, the approach will consume less power because it will refer each time to the prediction model, we believe that, in the long term, when a user's habits are well learned, the *PC* value of our approach will be stabilized under 7% , which will lead to about 60 % of power saving. To conclude the experimentations, we have tracked the phone's memory usage for each method for 168 h. Results presented in Table 2 represent the mean RAM usage of each application where our hybrid solution shows a promising RAM usage rate, better than the other solutions.

Table 2: Comparing our hybrid solution to our old solution, CB-SMOT and Waze application in terms of memory usage

	Our Hybrid solution	Our old solution	LifeMap
RAM usage	19Mo	35Mo	38Mo

#### 4. Conclusions and future works

In this paper we proposed a new battery-saving technique for extracting semantically and incrementally important geographical locations from users' moves. We learn users' habits to reduce the computational complexity of our approach, the proposed system is divided into three parts; the first part aims to recognize users' activities when they visit places for the first time, the second part is activity prediction where we predict the next activity to avoid processing the activities already recognized, and finally, activity verification which is a post-processing step that aims to verify if the predicted activity is the right one. This work is designed for detecting outdoor activities. However, supporting indoor activities represent a challenging future research direction that will let having an accurate information about the activity performed inside the POI.

#### 5. References

- [1] LifeMap Dataset. (2016), Mobility data collected by LifeMap monitoring system, Seoul. <http://www.crowdad.org/yonsei/lifemap/20120103/>
- [2] J. Chon, H. Cha, "LifeMap: A Smartphone-Based Context Provider for Location-Based Services", IEEE Pervasive Computing, vol.10, no. 2, pp. 58-67, April-June 2011, doi:10.1109/MPRV.2011.13.
- [3] L. Spinsanti, F. Celli, C. Renzo, where you stop is who you are: understanding people's activities by places visited. In the proceedings of Behaviour Monitoring and Interpretation (BMI) workshop, 2010.
- [4] Takeuchi, Y. et al. CityVoyager: An Outdoor Recommendation System Based on User Location History. In *Proceedings of UIC'2006*, (Berlin, 2006), Springer Press: 625-636.
- [5] M. Boukhechba, A. Bouzouane, B. Bouchard, C. Gouin-Vallerand, S. Giroux: Online prediction of people's next Point-of-Interest: Concept drift support. Proc. 6th international workshop on Human Behavior Understanding, UBICOMP 2015, Osaka.
- [6] M. Boukhechba, A. Bouzouane, B. Bouchard, C. Gouin-Vallerand, S. Giroux: Online recognition of people's activities from raw GPS data: Semantic Trajectory Data Analysis. Proc. The 8th ACM PETRAE Conference, July 2015, Greece.
- [7] Bezdek, J. C., and Ehrlich, R. W., FCM: The Fuzzy C-Means Clustering algorithm., Computers & Geosciences, Volume 10, Issues 2 3, Pages 191203, 1984.
- [8] LifeMap. (2016), Mobility Monitoring Tool: <http://sourceforge.net/projects/lifemapandroid/>.
- [9] Hipp, J.; Güntzer, U.; Nakhaeizadeh, G.: Algorithms for association rule mining: a general survey and comparison. ACM SIGKDD Explorations Newsletter 2: 58.
- [10] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng and H. Xiong. Discovering Urban Functional Zones Using Latent Activity Trajectories, IEEE Transactions on Knowledge and Data Engineering (TKDE), 2015, vol. 27, issue 3, 1041-4347.