Hindawi
Mobile Information Systems
Volume 2017, Article ID 3876906, 21 pages
https://doi.org/10.1155/2017/3876906

Research Article

Semantic Labeling of User Location Context Based on Phone Usage Features

Helena Leppäkoski,[1] Alejandro Rivero-Rodriguez,[1] Sakari Rautalin,[1] David Muñoz Martínez,[2] Jani Käppi,[3] Simo Ali-Löytty,[1] and Robert Piché[1]

[1]*Tampere University of Technology, Tampere, Finland*
[2]*GE Healthcare, Helsinki, Finland*
[3]*HD Automotive Positioning Solutions at HERE, Tampere, Finland*

Correspondence should be addressed to Helena Leppäkoski; helena.leppakoski@tut.fi

Received 16 February 2017; Revised 27 May 2017; Accepted 13 July 2017; Published 24 August 2017

Academic Editor: Antonio de la Oliva

In mobile phones, the awareness of the user's context allows services better tailored to the user's needs. We propose a machine learning based method for semantic labeling that utilizes phone usage features to detect the user's home, work, and other visited places. For place detection, we compare seven different classification methods. We organize the phone usage data based on periods of uninterrupted time that the user has been in a certain place. We consider three approaches to represent this data: *visits*, *places*, and *cumulative samples*. Our main contribution is semantic place labeling using a small set of privacy-preserving features and novel data representations suitable for resource constrained mobile devices. The contributions include (1) introduction of novel data representations including accumulation and averaging of the usage, (2) analysis of the effect of the data accumulation time on the accuracy of the place classification, (3) analysis of the confidence on the classification outcome, and (4) identification of the most relevant features obtained through feature selection methods. With a small set of privacy-preserving features and our data representations, we detect the user's home and work with probability of 90% or better, and in 3-class problem the overall classification accuracy was 89% or better.

1. Introduction

The use of smartphones has dramatically changed during the last decade. Whereas only 1% of worldwide population owned a smartphone in 2006 [1], now the number has reached 24% [2]. Mobile phones have become the most personal computing device. Users carry them continuously throughout the day and expect them to deliver meaningful services on the move. In order to provide a more personal and relevant user experience, mobile services can benefit from knowledge about the user's context. Context sensing can deliver new ways in how people interact with mobile devices by making the devices appear to be more human and personal. Intelligent devices can recognize the user, adapt to the user and the user's context, and learn to be proactive.

The most well-known context-aware applications are location-based services [3]. The location is usually represented by a set of coordinates defining a point or area on the Earth. This representation does not provide direct information about the meaning and relevance of a place to the user. Although in some locations it may be possible to use reverse geocoding to infer the type of the place, it is difficult to infer the meaning of the place for each user as the same place can have different meaning for different people. For example, a gas station might mean a frequent visited place, a work place, or just a nearby place during the daily commute. By leveraging the sensing capabilities of today's mobile phones, it is feasible to build a model that provides context related information about the user location.

This work aims to provide a reliable method to infer the meaning of the visited places of mobile phone users. We propose a machine learning based method for semantic labeling that utilizes phone usage features to detect the user's home, work, and other visited places. Our proposal provides better understanding of the user's location context and allows mobile phones to deliver more personalized and intelligent services and applications to users. For example, applications that are aware of the user's semantic location could allow the user to set reminders to phone to trigger when leaving home, arriving to work, or going to a frequently visited place, or to set automatic functions based on current place, for example, changing profiles or silencing phone.

In this work we develop a system to learn and label a user's places based on phone usage and analyze the effects of different choices of data representation. Our goal is an automatic method for detecting places of a user by applying a classification model learned from the data of the other users. This is similar to a use case where the earlier users of an application have contributed to the model by providing their data, and later, using the model, the application labels the data of the new users. Our contributions include (1) the introduction of novel data representations including accumulation and averaging of the usage data and performance results based on the proposed data representations, (2) analysis of the effect of the data accumulation time on the accuracy of the place classification, (3) analysis of the confidence on the classification outcome, and (4) identification of the most relevant features obtained through feature selection methods.

For training and model assessment we use two data sets. One of these is the Mobile Data Challenge (MDC) database [4, 5], where about 200 users used Nokia N95 devices normally for time spans between 3 and 18 months. The data includes logs of phone calls and SMS, calendar entries, multimedia displayed, GPS information when available, network information, and system information (e.g., battery status, device inactive time). The other data set is smaller: it covers a shorter time span (1–3 months) and includes labeled data of 16 users. This data includes information on similar phone usage and activity patterns as the MDC data, but there are differences in what is measured and how the observations are processed before storing them, which also makes the available features different. Using the aforementioned data, we use supervised learning methods to create a place detection algorithm that estimates the semantic label of the current place based on the phone's current usage features.

The rest of this article is organized as follows. In Section 2 we outline the background of our work, highlighting the current needs for place detection. In Section 3 we present the data and features used in this work. Section 4 describes the methods used in the analyses and comparisons in this work: the data preprocessing and the data representations, different classification methods, the cross-validation method used in the comparisons, feature selection methods, and finally methods for assessing the confidence in the classification result. Section 5 presents the results of the analyses and comparisons. In Section 6, we discuss the findings of this work and summarize its similarities and differences to the related work. Finally, in Section 7 we conclude the article.

2. Related Work

Research on context-aware systems began in earnest in the early 1990s [10]. Context can refer to any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object [11]. To infer a user's context, we use sensor information. Following Baldauf et al. [11], the notion of a sensor is generalized to encompass any data source. We distinguish three types of sensors.

Physical sensors are devices that detect and respond to input from the physical environment and capture physical data.

Virtual sensors capture contextual information from applications and services. They can be based on local services (e.g., calendar) or external services (e.g., weather forecast).

Logical sensors provide contextual information by combining information from physical and virtual sensors.

Most existing context-aware systems consider physical sensors [12], including the sensors related to the user's position, such as GPS, accelerometer, gyroscope (allowing, e.g., activity recognition) [13, 14], or sensors that measure the properties of the user's environment, such as magnetic field, light, or properties of various radio signals [15, 16]. Regarding virtual sensors, one of the most used sensors is the user's language. For instance, Google provides developers with the user's language through function `getDisplayLanguage` in the Android Developers API [17]. Other context related information can be provided to mobile applications in similar fashion.

Researchers have pointed out that, in addition to sensors, the usage of mobile phones can provide meaningful information about the user's context [6–8, 18]. Do and Gatica-Perez [18] assert that the user's context can be inferred based on the usage of applications (e.g., calls, e-mail, and web browser). Rahmati et al. use the smart phone's context information including time, day, movement information from accelerometer, cell ID location, and GPS location together with usage context (the prior visited web site, phone call, and application) to predict the next usage of the phone [19].

In this work we continue that line of research by studying the association between the usage of mobile phones and the user's context. More concretely, we investigate the main challenges and possible solutions for place detection, a particular case of semantic labeling. Place detection provides important information to improve context-aware applications.

Aiming at improving current place labeling techniques, we apply different supervised learning methods on mobile phone usage log data to find models that, based on the mobile phone usage patterns, allow assigning semantic labels to the places the user visits. The preliminary results of our work have been presented in [20]. This paper enhances the contributions of [20] in the following aspects: (1) we introduce here third data representation and cumulative samples; (2) in the analysis, we use two data sets instead of only one; (3) we provide results on the effect of the accumulation time of the cumulative samples on the accuracy of the classifier models that we use to provide the place labels; (4) we use sequential feature selection to decrease the computational load and

to improve the accuracy in the prediction phase when the classifier models are used to predict the place label; (5) we study the assessment of the confidence of the classification results; (6) we enhance the set of classification methods we used for the analysis to include also support vector machines and logistic regression.

Other works have been carried out with similar goals to ours [6–9, 21], that is, semantic place prediction, and use data derived from the same database as data set #1 in our work. They differ from our work in the following aspects: the number of features we used for our classification method is only 14 at most, while the other works use more features; we use different classifiers; while the other papers classify all the 10 labels available in data set #1, we prioritize recognizing *Home* and *Work* and therefore combine all the less frequent labels to one label *Other*; and we present a comparison between three different data representation schemes: visits, places, and cumulative samples. We also show that the accuracy can be improved by selecting a subset of the most relevant features to be used in the classification model and we study the benefit of rejecting classification results that obtain low confidence ranking from the classifier. Since its publication, the MDC data set has been extensively used in research. In addition to the semantic place prediction, it has been used, for example, in research of mobility patterns of phone users [22–24] and in human mobility prediction (prediction of the next location) [25, 26].

The research presented in this paper was conducted as part of the related work for the creation of the Place Monitor API of the Lumia SensorCore SDK [27]. The SDK is a collection of APIs to provide meaningful activity and location data from sensors that run constantly in the background in a low power mode.

3. Description of Data Sets

We used two different data sets for learning and predicting semantic place labels. In this section we describe the data and identify the most relevant features for place detection.

3.1. Data Set #1. Data set #1 is obtained from the MDC database made available by Idiap Research Institute, Switzerland, and owned by Nokia [4, 5]. The data set contains Nokia N95 smart phones usage data, collected by nearly 200 users over time periods that for many users exceed one year [5]. The information about the usage of the phones was automatically collected and anonymized. After the data collection, a clustering algorithm was used to identify the most relevant places for each user, that is, places that the user visited often and spent lot of time. These places the users labeled manually [9]. As our main focus was in the detection of *Home* and *Work* where people usually stay longer times, we extracted for our tests the data that was collected during the visits where the user stayed at least 20 minutes in the same place. The time intervals of these visits are defined in a database table `visits_20min.csv`, which is included in the MDC database, and it defines the start and end times, user ID, and place ID for more than 55,000 visits (see Figure 1). The place labels for the place IDs are defined in a separate

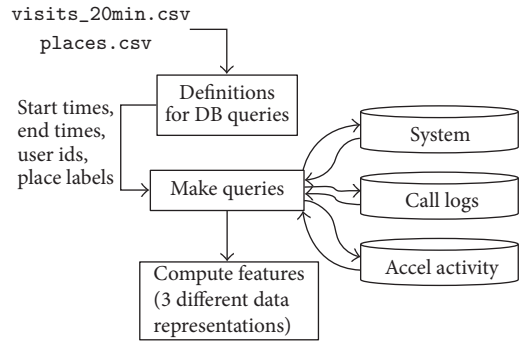

Figure 1: Obtaining data for feature computation.

MDC database table `places.csv`. The place in the database is defined so that it corresponds to a circle with 100 m radius [5].

Based on these data, we queried from the database the following phone usage data for each visit, that is, for a given user, all data entries between the start and end times of the visit:

(i) System data, including battery and charging status and counter for inactive time

(ii) Call log, including durations of each phone call

(iii) Acceleration based activity data, including accelerometer based estimates of the user's motion mode: *idle/still, walk, car/bus/motorbike, train/metro/tram, run, bicycle*, or *skateboard*. Due to the large area covered by a place, it is possible that the data from one place contains also significant amount of mobility, for example, walking or even being in a moving vehicle.

From these data entries, we computed for each visit the features to be used in the classification task. We decided to use only such sensor data that can be assumed to be available also for a real time application on a phone without violating the privacy of the user. Our feature list includes the following:

(i) duration: duration of the visit in seconds

(ii) startHour: time of the day when the visit started $(0, 1, \ldots, 23)$

(iii) endHour: time of the day when the visit ended $(0, 1, \ldots, 23)$

(iv) nightStay: proportion of the visit duration that is between 6 pm and 6 am

(v) batteryAvg: average battery level

(vi) chargingTimeRatio: proportion of the visit duration when the charging has been on

(vii) sysActiveRatio: proportion of the visit duration when the system has been active

(viii) sysActStartsPerHour: number of status changes from system inactive to system active divided by the visit duration in hours.

For features related to calls, both incoming and outgoing voice calls are taken into account:

(i) callsTimeRatio: the ratio of accumulated duration of calls to the duration of the visit

(ii) callsPerHour: number of calls divided by the visit duration in hours.

The features related to accelerometer based motion mode detection were computed using the reported motion modes. However, as the report for one time instance may include several different modes and includes also their probabilities, we used the probabilities to weight the times for the motion modes:

(i) idleStillRatio: proportion of the visit duration when the status is *idle/still*

(ii) walkRatio: proportion of the visit duration when the status is *walk*

(iii) vehicleRatio: proportion of the visit duration when the status is *car/bus/motorbike* or *train/metro/tram*

(iv) sportRatio: proportion of the visit duration when the status is *run, bicycle,* or *skateboard.*

In addition to these 14 calculated features, we also saved the place label and user ID to be used in the training and testing of the models:

(i) placeLabel: three possible labels: *Home*, *Work*, or *Other* (the last includes all the generally less frequent places, such as friend's home, transportation, and restaurant)

(ii) userId: each data sample includes a unique user identifier.

The MDC data includes place labels that were provided by users [9]. First, the data were collected and the relevant places for each user were clustered. In a later stage, users were shown all the places on a map and were asked to label these places. We only consider places labeled with certainty and left out those places that users were not sure about or did not label.

In total, the visits data includes 55,932 labeled visits by 114 distinct users. From the visits, 28,921 instances are to Home (52% of all visits), 21,697 instances to Work (38%), and 5,314 instances to Other places (10%).

3.2. Data Set #2. Data set #2 was provided by Microsoft and collected by 16 users working in the ICT field. The average time the participants collected data was 26 days and the maximum time was 64 days. The description and results on this data set have not been published earlier.

In this data set, the data was associated with places. The place was identified by its physical location, obtained, for example, from the GNSS receiver or cellular network based positioning. The first time the user visited a place, a new data entry for that place was created. Every time the user visited a once created place, the phone accumulated time counters for several status variables. The *stay time* was the accumulated time the phone was observed to be in the place and *night stay* was the accumulated time the phone was there between 6 p.m. and 6 a.m.

The accumulated times included also the times with the motion states *idle, stationary, moving, walking,* and *vehicle*, all determined by the sensors of the phone. The third group of times recorded included phone usage data: time with *display on* and *charging* times, time spent on *calling*, and time with *headset on*.

In addition to these, the *total time* since the place data entry was created was recorded. To the place data, the user-given semantic label, such as *Home* or *Work* was also associated. The physical location of the place was not included in the data. Twice a day, when a data connection was possible, the phone application sent the recorded time countervalues to a server. Thus the database included the history of the countervalues that were sampled at approximately 12 h intervals. This data set differs significantly from data set #1 in that the individual visits to a place cannot be detected or counted and neither can the individual phone calls or activity starts. From this data, we computed the features to be used in classification by dividing the time countervalues by the total time. Similarly, as with data set #1, we lumped all other user-given place labels, except *Home* and *Work*, to the third label *Other*.

In total, data set #2 includes 5,605 labeled samples by 16 distinct users. From these samples, 1,747 cases (31% of all visits) are labeled with *Home*, 1,482 with *Work* (26%), and 2,376 with *Other* (42%). Each sample consists of 11 features related to stay, activities, and phone usage and additional information such as user id, place label, and total time recorded for the location.

With both data sets #1 and #2, regarding the accelerometer based recognition of the motion state or activity, we rely on the output of the motion or activity recognition functions of the phone applications and data set providers. The reliability statistics of the functions are not known to us. For our classification functions, the possible errors in these features are noise in the data.

4. Methods

We consider three alternatives for the data representation: visits data representation, places data representation, and cumulative samples; these terms are explained in Section 4.1. Once the data is extracted from the database in the representation schemas, we apply seven well-known classification methods. Our goal is to determine which classification method and which data representation approach is the best for the semantic labeling of places. We also describe the cross-validation method we used to assess the performance of the classification, the sequential feature selection method used to improve the accuracy and to assess the significance of the individual features, and the approach used for assessing the confidence in the classification results.

4.1. Data Representations. In this paper, we consider three different approaches to represent the data. The visits approach uses the features computed for each visit as such, so that the data includes several samples of one user's visits to each of the user's places. That means that there is one tuple for each location-user-event. Therefore, a user visiting home 3 times adds three tuples to the learning data. From data set #1, we extract 55,932 labeled visits by 114 users.

The places approach combines all the visits of one user to one place into a single summarized sample. That means that there is one tuple for each location-user, which is calculated combining all the relevant visit tuples. The idea is to assume that different users tend to use their phones in similar ways in semantically similar places, for instance, at home. From data set #1, we extract 295 labeled places by 114 users. For instance, if a user visited home ten times in a week, the visit data representation creates ten different data instances,

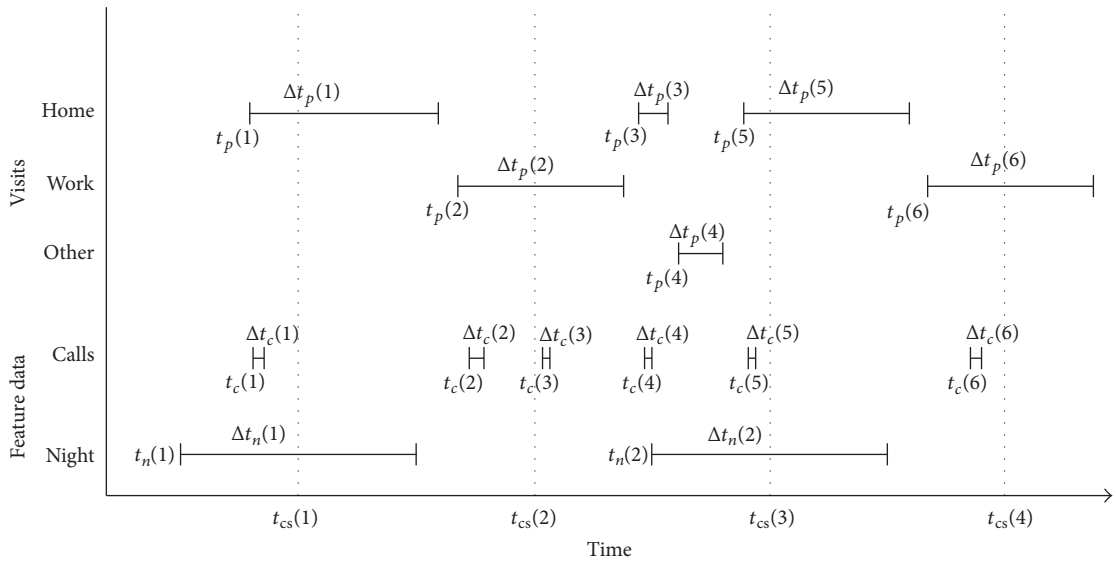

Figure 2: Example: time tags, durations, and place labels for computing features related to phone calls and night stay. The time of the jth cumulative sample is $t_{cs}(j)$.

while the place data representation combines the ten visit data instances into one place data instance. The visits and places representations are available only for data set #1.

The third data representation includes cumulative samples of the features. This representation is the native representation of data set #2; that is, it includes the accumulated times of staying and phone usage in a place by one user. To obtain similar samples from data set #1, we computed the accumulated times of stay, activities, and phone usage for each user-place combination. We took samples of these accumulated times at 12 h intervals and divided them by the total times since the first sample of the user-place combination. The 9 features of data set #1 that were converted to cumulative samples are the following: *stay, nightStay, charging, sysActive, calling, idleStill, walking, vehicle*, and *sport*.

Figure 2 and Algorithm 1 illustrate the computation of the different data representations for features related to phone calls and night stay. Feature *nightStay* is chosen as an example because it is computed differently from all the other features and therefore needs to be described separately. By contrast, feature *calling* is similar to all other features, and the description of its computation can be applied also to these. Figure 2 illustrates the notation for the time tags and durations. The start time and duration of visit i are $t_p(i)$ and $\Delta t_p(i)$, respectively. In addition to these time attributes, the label place(i) is attached to the visit data. Similarly, $t_c(i)$ and $\Delta t_c(i)$ represent the start time and duration of a call, and $t_n(i)$ and $\Delta t_n(i)$ are the start and duration of night. In our implementation, $\Delta t_n(i)$ is constant 12 h. We also make the simplification regarding calls that span over a visit so that the whole call is associated with the visit where it started.

Algorithm 1 presents equations for the computation of the features for the different data representations. As examples of combining data from several visits to places and cumulative samples data representations, we use *Home* as the example of user's place. The feature for number of phone calls is not included in cumulative samples and, therefore, it is not included in Algorithm 1. On the other hand, *nightStay* is included only in the feature set of cumulative samples. Although a call can take place during the night, in the data representation *nightStay* and *calling* are not directly connected. However, for the classifier model it is possible to learn the connection as their simultaneous occurrence increases both counters *(calling*, *nightStay)* simultaneously.

4.2. Classification Methods. In this work we apply the following classification methods using their implementation in the Statistics and Neural Networks toolboxes of Matlab.

Naïve Bayes (NB) [28–31] is a statistical approach having an explicit underlying probability model and it provides a probability of being in each class rather than simply a classification. Naïve Bayes assumes that features are conditionally independent; this reduces computational cost and often works well even if the independence assumption does not hold. There are no tuning parameters in this approach.

Decision tree (DT) [28–31] is a machine learning approach that probably gives the most understandable results by humans, who can identify the most relevant features. For attribute selection we use Gini's diversity index. The features selected at the top of the tree are the most relevant features for the classification. There are two options to avoid overfitting, prepruning, and postpruning. We chose postpruning since prepruning requires determining when to stop growing the tree while building it, which is not an easy task. When the tree is built we postprune the tree using Error Estimation. Intuitively, the method goes through the nodes of the tree comparing the original tree with the tree pruned on that node. The tree is pruned in that node if the pruned tree improves (or equals) the classification accuracy.

Assume that start times and durations are available for
(i) visits: $t_p(1), t_p(2), \ldots$ and $\Delta t_p(1), \Delta t_p(2), \ldots$
(ii) calls: $t_c(1), t_c(2), \ldots$ and $\Delta t_c(1), \Delta t_c(2), \ldots$
(iii) nights: $t_n(1), t_n(2), \ldots$ and $\Delta t_n(1), \Delta t_n(2), \ldots$
visit i (compute for all i)
(i) find the smallest index k_1 such that $t_p(i) \leq t_c(k_1)$
(ii) find the largest index k_2 such that $t_c(k_2) \leq t_p(i) + \Delta t_p(i)$

$$f_{\text{callsTimeRatio}}(i) = \sum_{k=k_1}^{k_2} \frac{\Delta t_c(k)}{\Delta t_p(i)}$$

$$f_{\text{callsPerHour}}(i) = \frac{(k_2 - k_1 + 1)}{(\Delta t_p(i) c_{t2h})},$$

where multiplication with c_{t2h} converts the time units to hours
places: home
(i) find set H of all visit indices i such that **place**$(i) = Home$

$$f^H_{\text{callsTimeRatio}} = \frac{(\sum_{i \in H} f_{\text{callsTimeRatio}}(i) \Delta t_p(i))}{(\sum_{i \in H} \Delta t_p(i))}$$

$$f^H_{\text{callsPerHour}} = \frac{(\sum_{i \in H} f_{\text{callsPerHour}}(i) \Delta t_p(i))}{(\sum_{i \in H} \Delta t_p(i))}$$

cumulative sample j: **home** (compute for all j)
Computation of $f^H_{\text{calling}}(j)$ for calling at home:
(i) find set H of all visit indices i such that **place**$(i) = Home$
and $t_p(i) + \Delta t_p(i) \leq t_{\text{cs}}(j)$

$$f^H_{\text{calling}}(j) = \frac{(\sum_{i \in H} f_{\text{callsTimeRatio}}(i) \Delta t_p(i))}{(t_{\text{cs}}(j) - (t_p(\min_{i \in H} i)))}$$

Computation of $f^H_{\text{nightStay}}(j)$ for night stay at home:
$a = 0$
for all $i \in H$
if exists k such that $t_n(k) \leq t_p(i) \leq t_n(k) + \Delta t_n(k)$

$$a = a + \min(t_p(i) + \Delta t_p(i), t_n(k) + \Delta t_n(k))$$

else if exists k such that $t_n(k) \leq t_p(i) + \Delta t_p(i) \leq t_n(k) + \Delta t_n(k)$

$$a = a + t_p(i) + \Delta t_p(i) - t_n(k)$$

end for

$$f^H_{\text{nightStay}}(j) = \frac{a}{(t_{\text{cs}}(j) - (t_p(\min_{i \in H} i)))}$$

Algorithm 1: Example: computing features related to phone calls and night stay in different data representations.

Bagged tree (BT) [29–32] combines different decision trees (with the same parameters as the decision tree above), each of which has been trained using different portions of the data. Using a voting system, each tree is given more weight in the region of the space where its classification rate is better. This method is proved to work better than single decision trees. We use ten decision trees, a typical value.

Neural network (NN) [28–31, 33] is a brain-physiology inspired classifier. It consists of layers of interconnected nodes, each node producing a nonlinear function of its input. The input to a node may come from other nodes or directly from the input data. Some nodes are identified with the output of the network. In particular, we used a multilayer perceptron with one hidden layer that contains ten hidden neurons. The decision of having these settings is based on the limited number of samples and the authors' experience. To train the network we used Levenberg-Marquardt optimization to update the weight and bias values. Neural network for classification assumes that the class labels are represented as binary vectors. Therefore, before training the class labels are coded as vectors: $Home \rightarrow [1, 0, 0]$, $Work \rightarrow [0, 1, 0]$, and $Other \rightarrow [0, 0, 1]$. The neural network predictions are also vectors. However, their element values are not exactly ones and zeros. The predicted classes are obtained by finding the index to the largest element of the output vectors and converting these back to class labels.

K-nearest neighbours (KNN) [28–32] is a statistical method that classifies an incoming instance according to the distance to the k-nearest points in the training set. We used Euclidean distance to choose the nearest neighbours. We determined the values of k to be used in classification using leave-one-user-out validation and classification accuracy as optimization criterion (see Section 4.4). We found that the best k value depends on data set and data representation: with data set #1 the best k values were 27, 3, and 57 for visits, places, and cumulative samples, respectively. With data set #2 the best accuracy was obtained with $k = 159$. For large training data sets, the required storage for the model is large, and

Table 1: Missing MDC data instances.

Data representation	Partial system data	Accelerometer based data	Both
Visits	192	36,543	25
Places	21	6	0
Cumulative samples	3,903	41,299	1,513

also the CPU time to find the nearest neighbours gets large. This may be prohibitive for applications running on resource constrained mobile devices.

Support vector machine (SVM) [30, 31, 33] is a binary classifier; that is, it can be applied for classification problems with two classes. A SVM seeks a hyperplane that best separates the features of one class from the features of another class. Its goal is to find a hyperplane that maximizes the zone on both sides of the hyperplane such that the zone does not include feature vector samples. The feature vectors closest to the found hyperplane are called support vectors. In many problems the separation of the classes cannot be done using a simple hyperplane. Therefore, the method includes a possibility of using linear or nonlinear kernel functions to produce a hypersurface that performs the separation. We used Gaussian Radial Basis Function (RBF) as the kernel function. With our data, we obtained similar accuracy with both the RBF and the linear kernel functions but RBF required smaller number of support vectors. We used Matlab's `fitcsvm` function to train the SVM classifiers. To set the RBF sigma parameter we used `KernelScale=1` which we found to work best with the data when compared with several other `KernelScale` values. The solution for our 3-class problem was obtained by using 3 binary classifiers to provide one-versus-all other classifications: *Home* versus *Not Home*, *Work* versus *No Work*, and *Other* versus *No Other*. For the binary classifiers, the multiclass labels were transformed before the training as follows: (1) *Home* $\rightarrow$ 1, *Work*, or *Other* $\rightarrow$ 0; (2) *Work* $\rightarrow$ 1, *Home*, or *Other* $\rightarrow$ 0; (3) *Other* $\rightarrow$ 1, *Home*, or *Work* $\rightarrow$ 0. In the prediction phase, the binary classifiers were used to obtain the posterior probabilities of their active class. The binary classifier with the largest posterior probability was used to determine the multiclass output.

Logistic regression (LR) [28, 31] models present the probability of the class as a logistic function of a linear regression expression of the features (linear combination of the features and a constant). LR is also a binary classification method. Therefore, we made a transformation of multiclass labels to several binary classes as we did in the case of SVM and trained three LR models. In the prediction phase the three classifiers were used to obtain the probabilities of the classes, and the class with the largest probability were chosen as the multiclass output. However, sometimes the linear regression problem is ill-conditioned and regularization is needed in order to obtain the parameter estimates. We used Lasso regularization for generalized linear model regression and constructed a regularized binomial regression model with 4 different values for regularization parameter λ and 2-fold cross validation. With these values the time consumed in parameter estimation remained moderate and the obtained model parameters provided good prediction accuracy.

4.3. Missing Data. With the MDC data, we encountered a problem with missing data. The data includes visits where either the system data partially (i.e., features *batteryAvg, chargingTimeRatio, sysActiveRatio,* and *sysActStartsPerHour*), the acceleration based activity data in full, or both of these data are missing. As the places and the cumulative samples representations are computed from the visits data, these representations inherit the problem. The numbers of instances with missing data in each of the data representations are shown in Table 1.

The instances with missing data cause problems in the training of the LR model and degrade the performance of other classifiers, especially NB, NN, and KNN. To mitigate the effect of missing data, we trained four variants for each classifier: the first one uses all the features; the second one uses all other features except the sometimes missing system features; the third one uses all other features except acceleration based ones, and the last one uses neither the sometimes missing system features nor the acceleration based features. The classifier variants were trained using only samples where all the features used by the classifier were available. In the evaluation of the classifier, the decision on which classifier variant to use for classification was made separately for each test data sample, we chose the classifier variant that did not require the features that were missing in the sample but used as many as possible of the features available.

4.4. Performance Evaluation of Classifiers. Once we have built the classifiers based on the training data, we use the test data to evaluate the classifiers. In machine learning, it is common to choose a certain proportion, for example, one-third, of data to a test set, which will be used only to evaluate the classifier, not to build the classifier model [29, 31]. The test set is also labeled. Therefore, we have the information about the true label (the user-given values) of the samples. In the evaluation of the classifiers, each test data sample is fed to the classifier and the output of the classifier, that is, the predicted label, is compared with the true label. Accuracy value of 53% means that 53% of the predicted values are equal to the true value; we use classification rate as a synonym of accuracy.

Our goal is to classify the data of one user by using a model based on the data of the other users; that is, we want to learn patterns that are common to all users. Therefore, splitting of the data to training and test sets is based on user id. As a result, the data of a user is not classified with the knowledge of the user's own data. Using knowledge of

future data of the user would be unrealistic, and using the knowledge of the past data of the user is a different problem, not addressed in this paper.

One option would be to randomly choose a certain proportion of users to test data. However, there is large variation in the numbers of samples by different users and the numbers of visits to each of the labeled places also differ by users. Because of this, the overall accuracy evaluations of the classifiers vary significantly depending on which users are in the test set. We solve this problem by using leave-one-user-out validation. For n users, the training and testing are repeated n times each time with one user's data as the test set and the remaining users' data as the training set. The overall accuracy obtained by combining the results from all the tests is used as evaluation criterion. This approach to cross-validation is deterministic, which makes the results easier to interpret when comparing several different setups, for example, in feature selection. In these comparisons, we want the variations in classifier designs, for example, the feature combination, to be the major sources of performance differences, not the random selection of test sets. The combined results include test results obtained using classifiers trained with all different training sets. It includes one classification result for each labeled data sample of each user. Note that we do not control the random initializations of training methods, which also makes some contribution to the observed differences. However, using leave-one-user-out validation and combining results of n tests mitigate also the biases caused by the random initializations.

4.5. Feature Selection. By selecting only a subset of the available features, the number of inputs presented to the classifier can be reduced. This benefits the classification task in several aspects: fewer features result in fewer model parameters, which improves the model's ability to generalize and reduce model complexity and the run time of the algorithm. It also provides insight into the problem by distinguishing the more significant features from the less important ones [32]. Some of the learning methods such as decision trees, bagged trees, and regularized logistic regression include feature selection as an integral part of the learning procedures [31]. However, others do not. Therefore, we search for the improved feature subset by selecting candidate subsets and evaluating their predictive accuracy using the leave-one-user-out validation described in Section 4.4.

One option for selecting subsets would be an exhaustive evaluation of all the possible subsets. However, for 11 features the number of subsets to be evaluated would be 2047 and for 14 features it would be 16,383. These would require too long computation time especially with slower methods, such as SVM, when applied for testing with leave-one-user-out validation. Therefore, a search strategy is needed for selecting candidate subsets for evaluation. We apply sequential selection algorithm for this purpose.

In sequential feature selection (SFS) features are added or removed one at a time [32]. The SFS provides a suboptimal solution to the feature selection problem as it easily becomes trapped to a local minimum. To mitigate this problem, we implemented the algorithm in both forward and backward directions. SFS in forward direction is a greedy search algorithm. It adds features one by one to the model until the addition of more features does not improve the predictive accuracy any more. In backward direction, the process is started from the model, including all the available features, and then features are removed one at a time until removing features does not improve the performance. Before the decision is made on which feature is added or removed, the effect of each available candidate feature for addition or removal is tested. The candidate feature that produces the largest improvement to the predictive accuracy when compared to the selected feature set from the previous trial cycle is added or removed, depending on the direction of the search. The process ends when none of the candidates in the entire trial cycle is able to improve the performance obtained in the previous trial cycle. If the predictive accuracy is the same as in the previous trial cycle, the candidate set with fewer features is selected.

4.6. Confidence of Classification. In many practical classification problems, it would be useful if, in addition to providing the classification result, the classifier was also able to provide information about the quality of its classification [34]. In particular, we focus on the confidence of the classification, assessing how reliable the classifier itself considers its own decision. High classification confidence means that the classifier is "sure" about its output while low confidence means it is "unsure." The idea in the confidence assessment is to use the information about the execution of the classifier on a specific input sample to infer the confidence that the classification result generated for the sample is correct [35].

NB and LR classifiers base their decisions on the probability models of the classes, and their output is the posterior probability of the class given the feature values. These probabilities can be considered as confidence measures of the classifier outputs. The SVM produces scores as class likelihood measures and Matlab provides `fitPosterior` function to transform these to posterior probabilities. The predicted outputs of the NN based classifier are binary vectors z of length $n = 3$, that is, the number of possible classes y. Ideally, the value of the element corresponding to the predicted class is 1 while the others are zeros. In practice, due to imperfect training examples, noise, and other mismodeled effects, the predicted elements z_y are seldom exactly ones and zeros. Therefore, the classification result $\hat{y}$ is determined using the element closest to one; that is,

$$\hat{y} = \underset{y}{\operatorname{argmin}}\, d_y, \tag{1}$$

where $d_y = |1 - z_y|$. Now the distance d_y serves as an indicator on how well the current feature vector fits to the NN model of class y. To get this value to the same scale with the probability outputs of NB, LR, and SVM, we convert the distance $d_{\hat{y}}$ to the confidence measure c. However, it may happen that, in some cases, when the fit of the input sample to the model is exceptionally poor, even the shortest distance $d_{\hat{y}}$ may be larger than one. Therefore, the confidence is obtained from the distance using $c = 1 - \min(1, d_{\hat{y}})$.

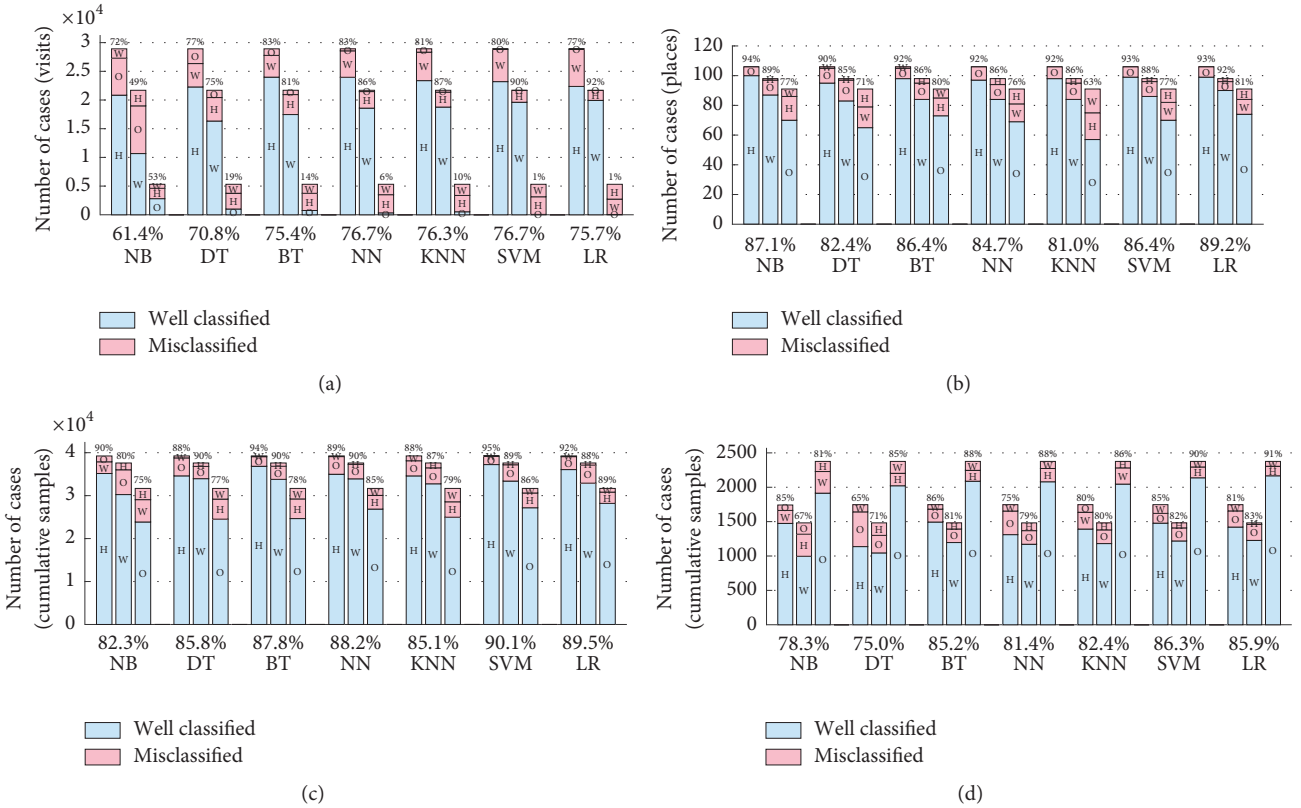

Figure 3: Classification rates (%) for different methods, using data set #1 and (a) visits, (b) places, and (c) cumulative samples approaches and (d) data set #2 and cumulative samples. The percentage of well-classified samples, that is, cases where the classification result is correct, for each class is given above the bars. The overall percentage of well-classified samples for the classifiers is shown below the bars.

Using both data sets, we study how well the confidence measure can predict the misclassification rate, that is, how well the classifier assesses its own performance. We set a threshold to the confidence, below which we say the confidence is low and above which it is high. There are four possible combinations of this measure (the confidence assessed by the classifier) and classification success: (1) well classified with high confidence, (2) well classified with low confidence, (3) misclassified with high confidence, and (4) misclassified with low confidence. The classifier produces the predicted label and confidence assessment based on the input features but without knowing the correct label. Therefore, it is possible that the classifier has high confidence but when its prediction is compared with the correct label, it turns out that the input was misclassified. We consider decisions 1 and 4 correct, as in these cases the confidence of the classifier predicts the success of the classifier, while in cases 2 and 3 the decisions are wrong as the confidence of the classifier gives wrong prediction about the success. Assuming that the costs of the unsuccessful cases 2 and 3 are equal, as well as the rewards of the successful cases 1 and 4 are equal, we search for a confidence threshold such that the ratio between the number of cases 1 and 4 over cases 2 and 3 is maximized. We use the obtained threshold to reject samples that have confidence lower than the threshold and record how much the overall accuracy of a classifier improves using the threshold and how large proportion of samples is rejected.

5. Results

In this section we describe our results on the comparisons of the data presentations and classification methods using the methods described in Section 4. In all the tests based on data set #1, the missing feature values in the input samples were treated as described in Section 4.3.

5.1. Classification. The results on the comparisons of the data representations and different classification methods are shown in Figure 3 where the evaluation criterion is the overall predictive accuracy observed in leave-one-user-out validation described in Section 4.4. The results are summarized in Table 2.

Figure 3(a) shows the classification of each method using the visits representation. All the methods but the Naïve Bayes show a certain bias. They achieve high accuracy for the places *Home* and *Work* and low accuracy for the place *Others*. The intuitive reason is that visits to *Home* or *Work* are more frequent than visits to places labeled as *Others*. Therefore, the algorithms sacrifice accuracy in *Others* to achieve higher accuracies in *Home* or *Work*.

Table 2: Accuracy results of the data representations: summary over all implemented classifiers.

	Data representation			
	a	b	c	d
min	61.4	81.0	82.3	75.0
max	76.9	89.2	90.1	86.3
mean	73.3	85.3	87.0	82.1
std	5.7	2.8	2.7	4.2
max-min	15.5	8.2	7.8	11.3

Figure 3(b) shows the corresponding results using the places representation. Compared to the visits representation, the classification accuracies are higher. Also, the differences between the accuracies of the classifiers are smaller than with the visits approach. The improvement obtained by combining of all the visits to one place may be because generally averaging reduces the effect of the outliers. The disadvantages of the places representation are the following. First, it is more computationally expensive to produce because of the need to combine all the individual visits to places. The second disadvantage is the so-called cold start problem: the classification algorithm will not classify accurately the places until a certain number of visits to a place have been collected.

The classification results with cumulative samples of data sets #1 and #2 are shown in Figures 3(c) and 3(d). The cumulative samples with data set #1 improve the accuracy and decrease the differences between the classifiers even more than the places approach. Cumulative samples include averaging similarly as the places representation and the generation of cumulative samples reduce variability in samples if the phone usage and place visiting pattern stay regular. However, the computation of cumulative samples also generates some variability, as it produces samples even when new visits have not been made to the place. In this case the feature values change as the total time used for scaling still grows even though the cumulative times of the stay and activities remain constant. The averaging together with the much larger number of samples provides a plausible explanation to the improvement. With the cumulative samples of data set #2, the accuracies are lower and the accuracy differences between the classifiers are larger. This could be due to the smaller size and time span of the data.

When comparing the results of different classifiers with all the data representations, SVM and LR are always among the three algorithms that provide the best classification accuracies while DT is among the three classifiers with the worst accuracy. BT and NN also perform quite well; they are never in the group of the worst three. Generally NB does not provide good accuracy, except that with the places representation it is the second in accuracy. From the classifiers studied in this paper, SVM is by far the slowest classifier to train. The classification with the trained SVM is fast; however, its memory requirements in classification phase become high if the number of support vectors is high. The issue is emphasized in multiclass classification as the support vectors need to be stored for each class separately. Therefore, despite its accuracy, SVM mainly serves as a reference, and we do not consider it to be suitable for practical applications with this type and amount of data in resource constrained mobile devices. The computational cost in prediction is also high with KNN as it has to store all the training samples and compare them with the new input. Therefore, its practical applications are restricted to cases where extreme simplicity of the algorithm is required but high computational costs can be accepted. Based on these comparisons, LR, NN, and BT seem to be the most promising methods for practical applications.

Our test results indicate that data representations including averaging, that is, places and cumulative samples, give higher classification accuracies than visits data representation. The average classification accuracies with visits, places and cumulative samples obtained from data set #1 were 0.72, 0.85, and 0.87, respectively, and 0.81 with the cumulative samples of data set #2.

5.2. Effect of Accumulation Time with Cumulative Samples. With the cumulative samples, the samples themselves evolve in time as new data are accumulated to the time counters of the features. To study the effect of the accumulation time to the classification accuracy, we grouped the samples based on the accumulation time t_{acc}. The first group included the samples where $t_{\text{acc}} \leq 1$ day, in the second group, was the samples with $t_{\text{acc}} \leq 2$ days and so on, until 7 days. These seven groups include the samples from the first week the user starts to visit a place. Into the eighth group we included all the samples, which gives the same classification accuracy that is illustrated in Figures 3(c) and 3(d).

In the training of the classifiers we used all the cumulative samples of all other users, so that the time based selection of samples did not affect the training phase. The results for the cumulative samples representation of data sets #1 and #2 are shown in Figures 4 and 5, respectively. In the figures, in addition to the overall classification accuracies, also the classification accuracies of the specific labels (*Home*, *Work*, and *Other*) are shown.

Comparing the overall classification accuracy of cumulative samples in Figure 4(a) and visits representation in Figure 3(a), it can be observed that after 6 days of accumulation time, the accuracy with cumulative samples is equal to or better than the accuracy with visits with all classifiers except NN and KNN. With these two the accuracy with visits were 76.7% and 76.3% while with cumulative samples and 6 days of accumulation time the accuracies are only about 72%. In Figure 4, the curves corresponding to the overall

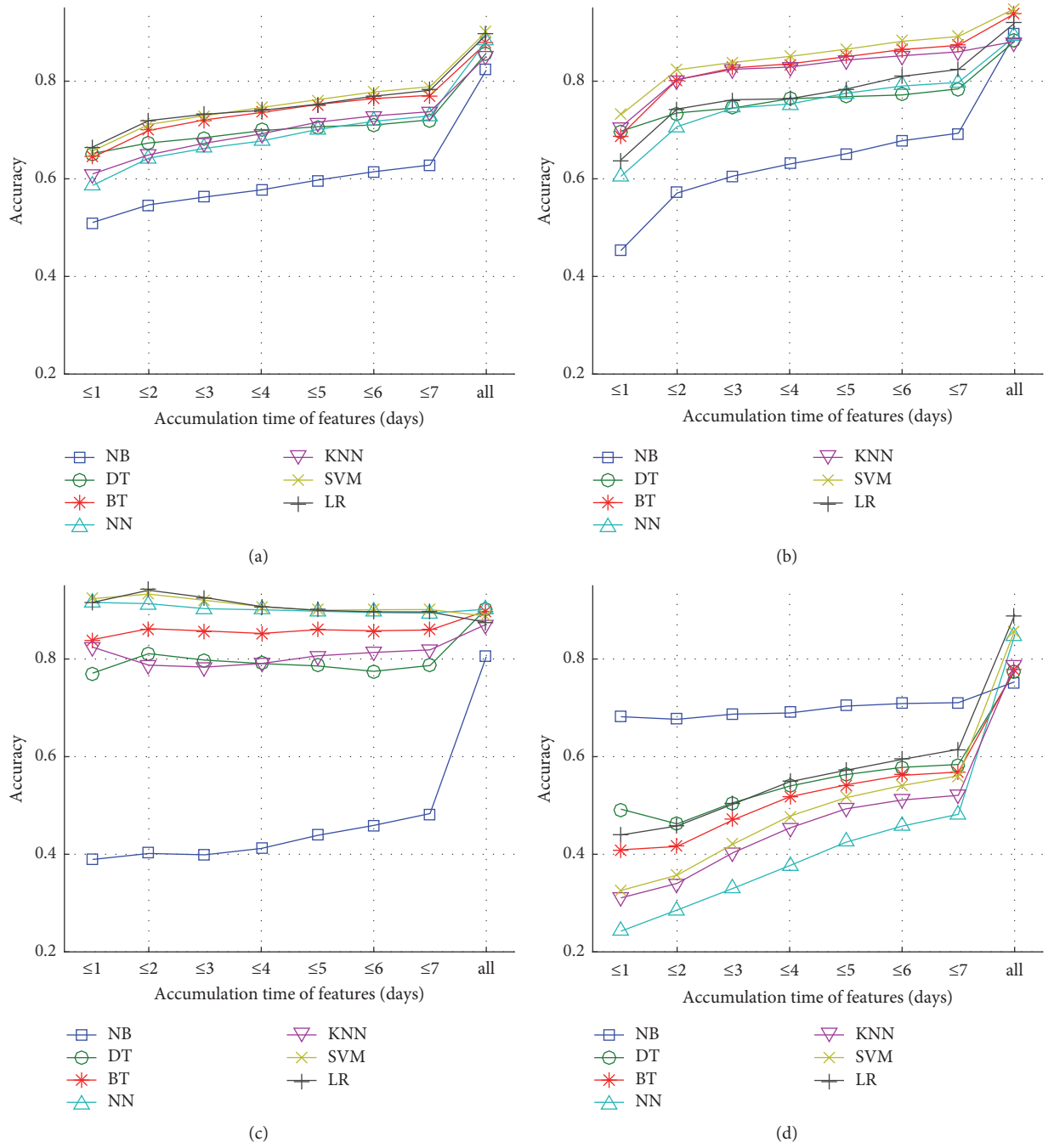

Figure 4: Effect of the accumulation time to the classification accuracy with cumulative samples of data set #1: (a) overall accuracy and classification accuracy of (b) Home, (c) Work, and (d) Other places. Accuracy as ratio (unitless).

accuracy and classification accuracy of *Home* and *Other* for all the classifiers are monotonically rising after 2 days; that is, the accuracy improves as the accumulation time of feature samples increases. There is also clear improvement from 7 days to the maximum accumulation time. The classification accuracy of *Work* behaves differently: with all classifiers except NB the rise of the accuracy is very slow and it is not monotonically rising.

Based on these results, for *Work* gathering more information by integrating the values for longer time does not improve its accuracy as happens with *Home* and *Other*. The data sets differ in that with #1 there is clear accuracy improvement when accumulation time increases from 7 days, while with #2 there is no clear improvement; with *Home* even a decrease of the accuracy can be observed. This is probably due to the smaller total number of samples and

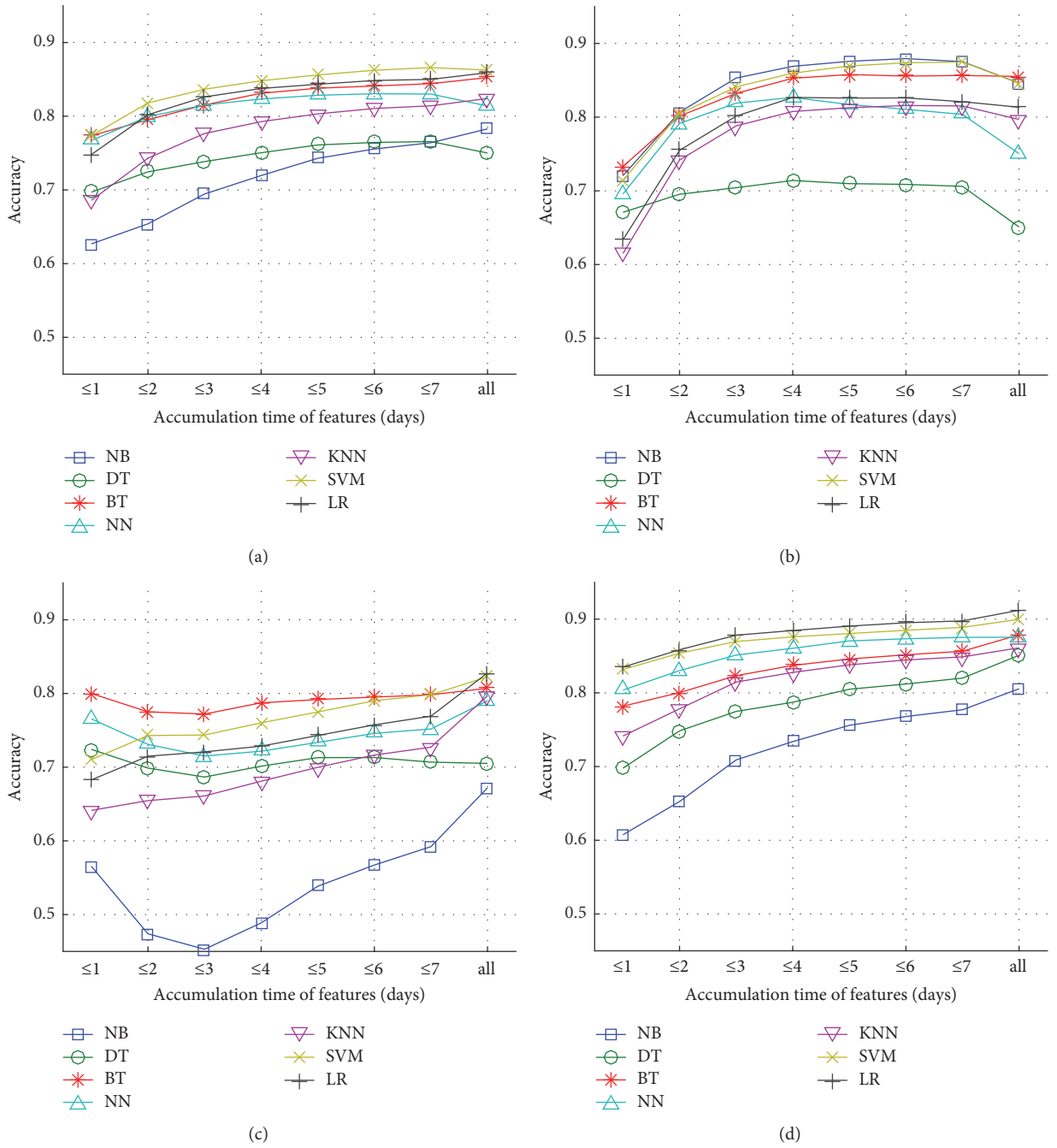

Figure 5: Effect of the accumulation time to the classification accuracy with cumulative samples of data set #2: (a) overall accuracy and classification accuracy of (b) Home, (c) Work, and (d) Other places. Accuracy as ratio (unitless).

shorter data collection times in data set #2. The histograms of accumulation times of the samples in both data sets are shown in Figure 6. With data set #2, about half of the samples have accumulation time less than 7 days. With longer accumulation times, the data is biased by only few users, which reduces the reliability of results on longer accumulation times.

In Figure 5(a) the overall accuracy approaches the final accuracy already after 4-5 days accumulation: only NB improves significantly; after that, BT, KNN, and LR improve only slightly, and the accuracies of DT and NN decrease. Comparing different data sets, the cumulative samples in Figure 5(a), and the visits representation of data set #1 in Figure 3(a), it can be seen that already after 2 days of data accumulation the accuracies with cumulative samples exceed the accuracies of visits. In Figure 5, only the classification accuracy of *Other* is monotonically rising for all the classifiers. In the accuracy of *Home* there is a clear drop from 7 days to the maximum accumulation time with all classifiers except BT, and with DT, NN, and LR the decrease starts even earlier

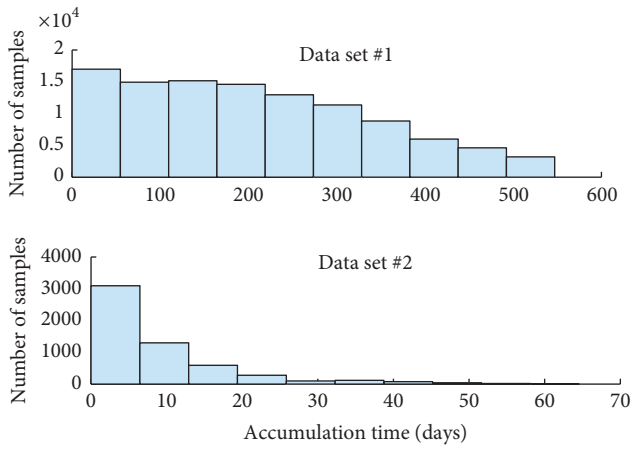

Figure 6: Histograms of the accumulation times with both data sets.

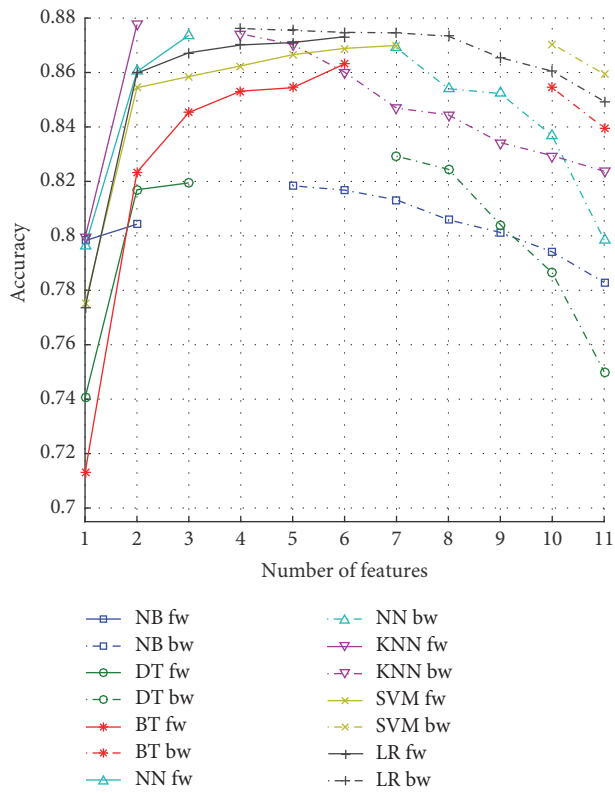

Figure 7: Sequential feature selection with several classification methods, in forward and backward direction. Accuracy as ratio (unitless).

before accumulation time of 7 days. In the classification of *Work* the accuracy behavior differs from other classes: with all other classifiers except SVM and LR, the accuracy first decreases with accumulation time and then starts to increase. With DT, the final accuracy is even worse than in the beginning. However, the increase of accuracy is very slow, except with NB. In spite of these effects in the classification accuracy of individual classes, in the first 7 days the overall accuracies shown in Figure 5(a) increase as the accumulation time increases. However, the accuracy with the maximum accumulation time with DT and NN is smaller than with 7 days of accumulation.

Generally, the longer time the data has been accumulated, the more accurately the data sample will be classified. The average accuracy obtained using the visits representation of data set #1 is exceeded by cumulative samples of data set #1 after 6 days of accumulation while with data set #2 that happens already after 2 days of accumulation.

5.3. Feature Selection. Sequential feature selection (SFS) in both forward and backward directions for all the classifiers described in Section 4.2 was applied to data set #2. The results are shown in Figure 7, where the overall accuracy of the classifier is shown as a function of the number of features. The curves with solid line show the results of SFS in the forward direction. For each classifier, the line starts from the left with one feature and continues until the addition of new features does not improve the accuracy any more. The results of SFS in the backward direction are shown with dash-dot lines. These curves start from the right with all 11 features included and continue to the left decreasing the number of features until removing features does not improve the results any more. The accuracies with just one optimally chosen feature are between 0.69 and 0.79 while with all features the accuracies are between 0.74 and 0.86. The accuracies using the best feature subsets found with forward and backward algorithms are between 0.82 and 0.87. Thus the selection of the features decreases the accuracy differences between the classifiers.

With NN, BT, and KNN, the forward selection yields better accuracy than backward selection and the number of selected features is also smaller. With NB, DT, and LR, the obtained accuracy in backward direction is better. With LR, the number of selected features in the backward direction is also smaller than in the forward direction, while with NB and DT better accuracy is obtained using more features than those selected by forward SFS. Using SVM, the best accuracies in both directions are approximately the same. However, in the forward direction only 7 features are needed while in the backward direction 10 features are required for the same accuracy. The three best accuracies are obtained using LR with 4 features, NN with 3 features, and SVM with 7 features. Interestingly, the accuracy using NN with just one optimally selected feature is approximately the same as with NN with all 11 features included.

The evolution of the feature subset composition during the forward and backward SFS is shown in Figure 8. The features selected in forward selection are shown in Figure 8(a): the bigger the weight and the size of the squares were, the earlier the corresponding feature was selected. The features that did not get selected at all are not marked with squares. Figure 8(b) shows the feature removals performed in the backward selection. The large dark squares show the features that were not removed during the selection process. The smaller and lighter the square was, the earlier the feature was removed; if the size and weight are reduced, the feature is not included in the final subset. Note that, in Figure 8, the

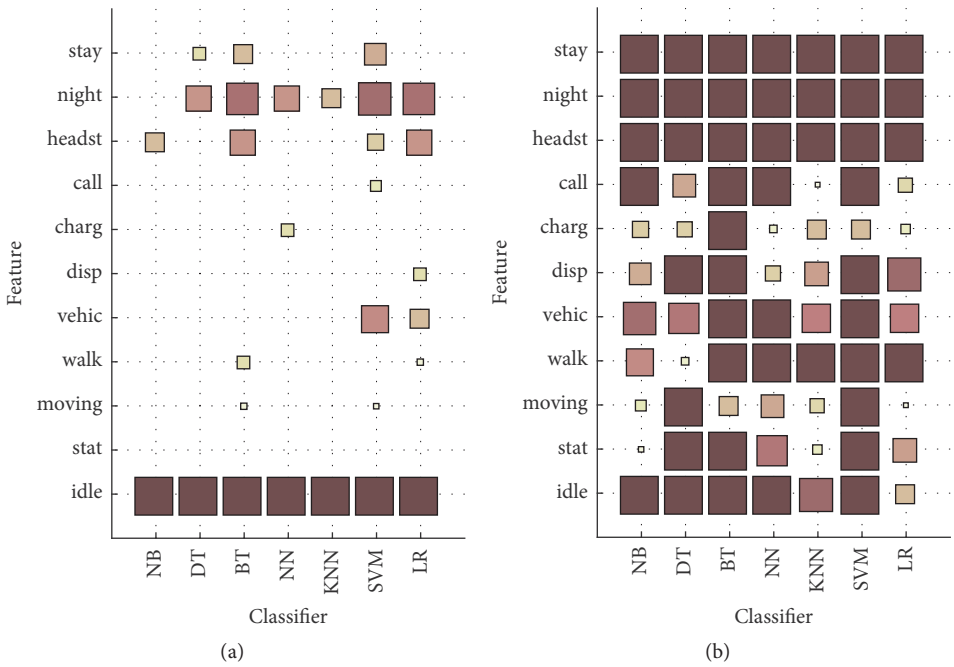

Figure 8: Importance of the features, based on sequential feature selection in (a) forward direction and (b) backward direction.

size and weight scales of the squares are comparable only within classifiers with the same final number of features. No feature is included in all final subsets, covering all classifiers and both directions of SFS. In forward direction, *idle* was the first feature selected into the model with all classifiers, and *nightStay* was the second feature selected with all other classifiers except NB. In backward direction, *stay*, *nightStay*, and *headSet* are included in all the final subsets and *idle* is included in final subsets of all classifiers except LR.

Based on these tests, we see that, even with the same training and test sets, the relevance of the features depends on the classifier. However, features *stay*, *nightStay*, *headSet*, and *idle* seem to be relevant for most of the classifiers. The selected feature sets provided improvements to the overall accuracy in the range 0.02–0.07, resulting in accuracies in the range 0.82–0.88. It can be noted that the accuracy of also the classifier models that inherently perform feature selection or extraction in their training phase, that is, DT, BT, and LR in our tests, can be improved using external feature selection algorithm. However, the results in feature importance are considered only as preliminary, as the small size of data set #2 reduces the reliability of these results.

With NB, DT, and SVM, the subset selected in forward direction is included into the final subset obtained in backward selection. With BT and NN, the features that were last selected in forward direction were first removed in backward direction and with LR the feature that was first selected in forward direction was the fourth feature removed in backward direction. This suggests that with this data, combining both the forward and backward selection in the SFS algorithm could improve the selected feature subset when accuracy is used as the selection criterion.

5.4. Confidence of Classification. To evaluate the relation between accuracy and the confidence measures defined in Section 4.6, we collected all the classification results and their confidence values that were obtained using test data and NB, NN, SVM, and LR classifiers. We ordered the results based on the confidence measure and divided them into 20 equal sized groups. For each of the confidence groups, we calculated the overall classification accuracies. The accuracies of these groups are shown in Figure 9 for both data sets and all the data representations.

With all the data representations, it is clear that the accuracy is significantly lower in groups with lower confidence value. However, even these groups include also well-classified samples. In the results in Figure 9(b), obtained with data set #1 and the places approach, the curves include many spikes. This is a quantization effect due to the small total number of samples. In general the curves in Figure 9(a) are smoother than in Figures 9(b)–9(d). Also the curves in Figure 9(a) show a more steady rise when compared to curves in Figures 9(b)–9(d) which present saturation-like behavior. One possible reason to the difference is the filtering that has been applied to the samples in Figure 9(b) by averaging the visits data and in Figures 9(c) and 9(d) by integrating the raw data.

In Figure 10 the ratios between the correct and wrong decisions of the classifiers are shown as a function of the confidence threshold. The threshold was used to reject classification results with confidence lower than the threshold. Correct decisions included the cases where the sample was classified correctly with confidence equal to or higher than the threshold or it was misclassified with confidence lower

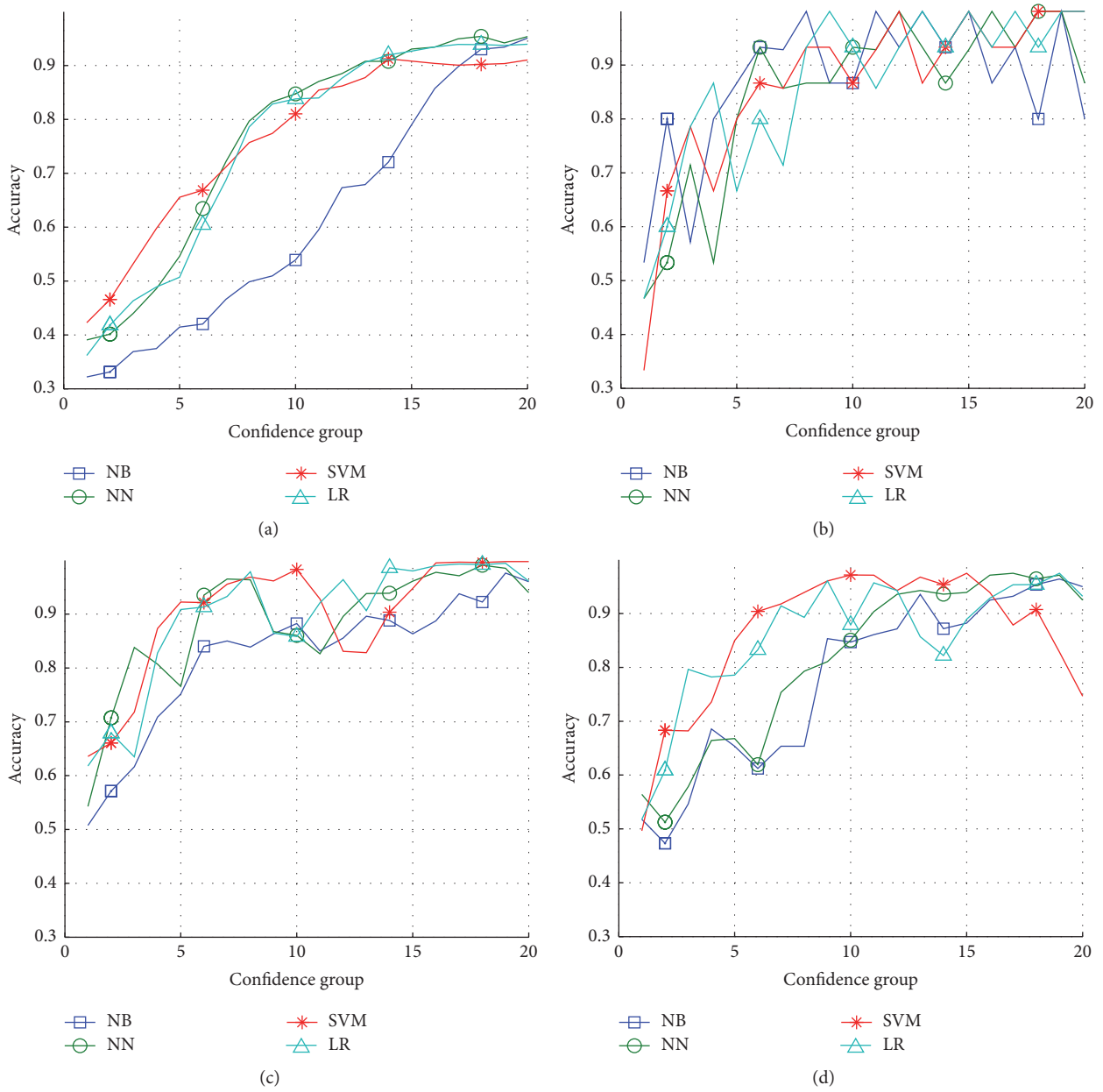

Figure 9: Accuracy (as ratio, unitless) in sample groups based on increasing confidence of classification for different data representations: data set #1, (a) visits, (b) places, and (c) cumulative samples; (d) data set #2, cumulative samples.

than the threshold. Wrong decisions included the cases well-classified with low confidence or misclassified with high confidence. The curves in Figure 10(a) are concave and smooth and include also parts where the curve is rising, making it easy to find maximums in the middle parts of the curves. In Figures 10(b) and 10(d) there are no clearly rising parts in the curves and in Figure 10(b) the curves are again wrinkled similarly as in Figure 9(b). The curves of LR in Figures 10(c) and 10(d) and SVM in Figure 10(c) are monotonically decreasing; that is, they have their maximums with the smallest confidence threshold.

Figure 11 illustrates the effect of the confidence threshold that maximizes the ratio between the numbers of correct and wrong decisions when the threshold is used to reject classification results with low confidence. Shown in the figures are the values of the confidence thresholds, the proportion of the samples rejected based on the threshold to the number of all samples, the absolute improvement of the predictive accuracy obtained by using the threshold, and the classification accuracy within the samples that are not rejected. In Figure 11(a) presenting the results of data set #1 and visits data representation, the rejection of results with lower confidence produce accuracy improvements varying between 0.05 and 0.14. With data set #1 and places data representation, shown in Figure 11(b), the improvements are clearly smaller, varying between 0.01 and 0.03. With

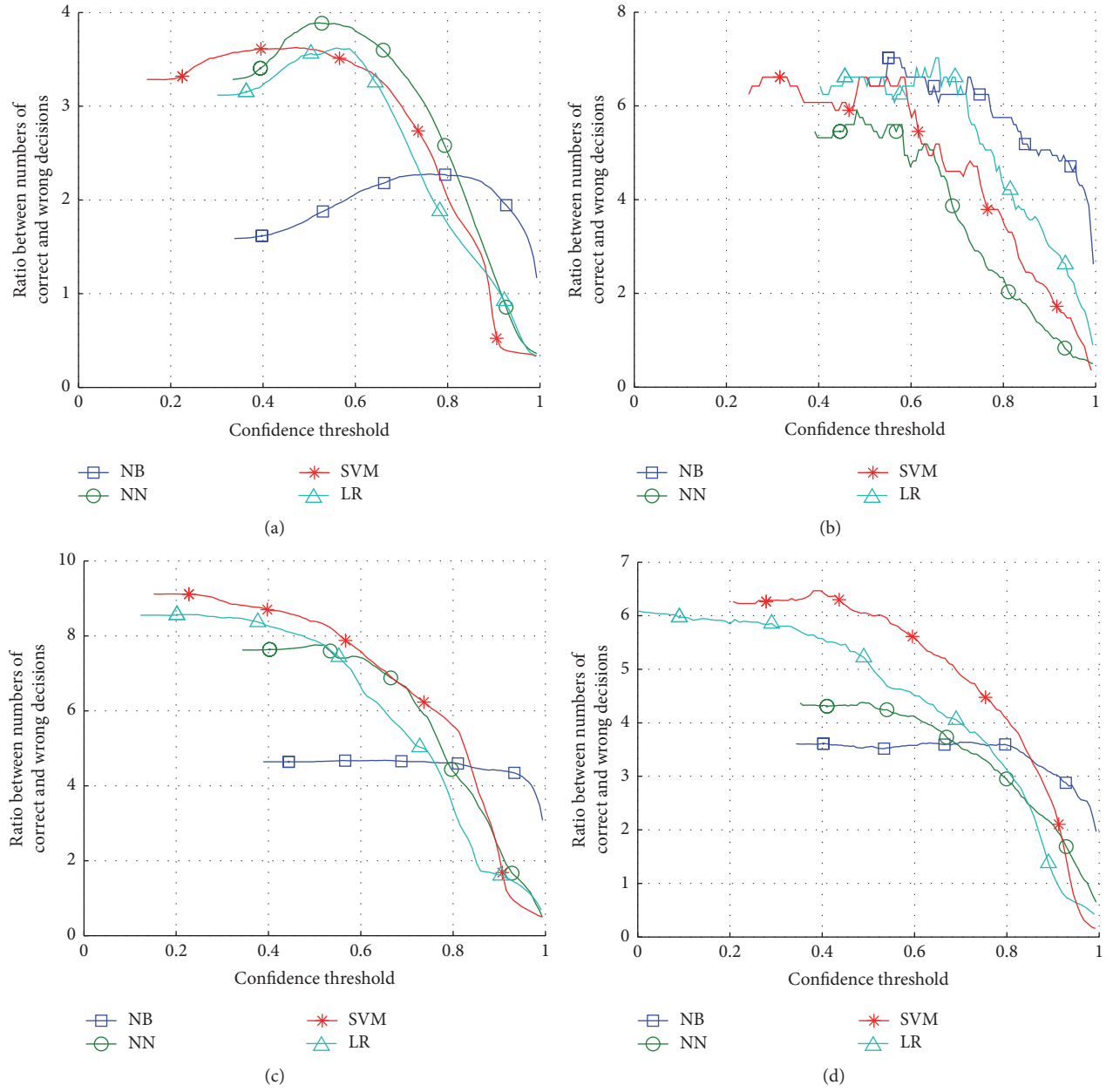

Figure 10: Ratio between the numbers of correct and wrong decisions as a function of confidence threshold. Data representations: data set #1, (a) visits, (b) places, and (c) cumulative samples; (d) data set #2, cumulative samples.

cumulative samples of the both data sets the threshold for LR rejects very few samples and the accuracy does not improve, as can be seen in Figures 11(c) and 11(d). With these data representations the improvements by the other classifiers are not significant either; with NB in Figure 11(d) the increase is about 0.03; in other cases it is about 0.01 or less. To summarize, with visits, the improvement obtained using confidence thresholds is more significant than with other data representations. However, even when applying thresholds, the accuracies are not as high as with places (compare the A bars of Figures 11(a) and 11(b)), but the difference is greatly reduced from Figures 3(a) and 3(b).

Comparing Figures 9 and 11, we see that the groups in Figure 9 with lower confidence and low accuracies, say below 0.5, have potential for accuracy improvements by rejecting results with low confidence, and the improvements are visible in Figure 11. However, based on these tests, with the data representations including averaging, the improvements are not significant.

In the results shown in Figures 9–11, also the determination of the thresholds is based on test data. Therefore, the effect of the threshold is not evaluated using independent data and, despite the modest improvements, these results may still be overly optimistic.

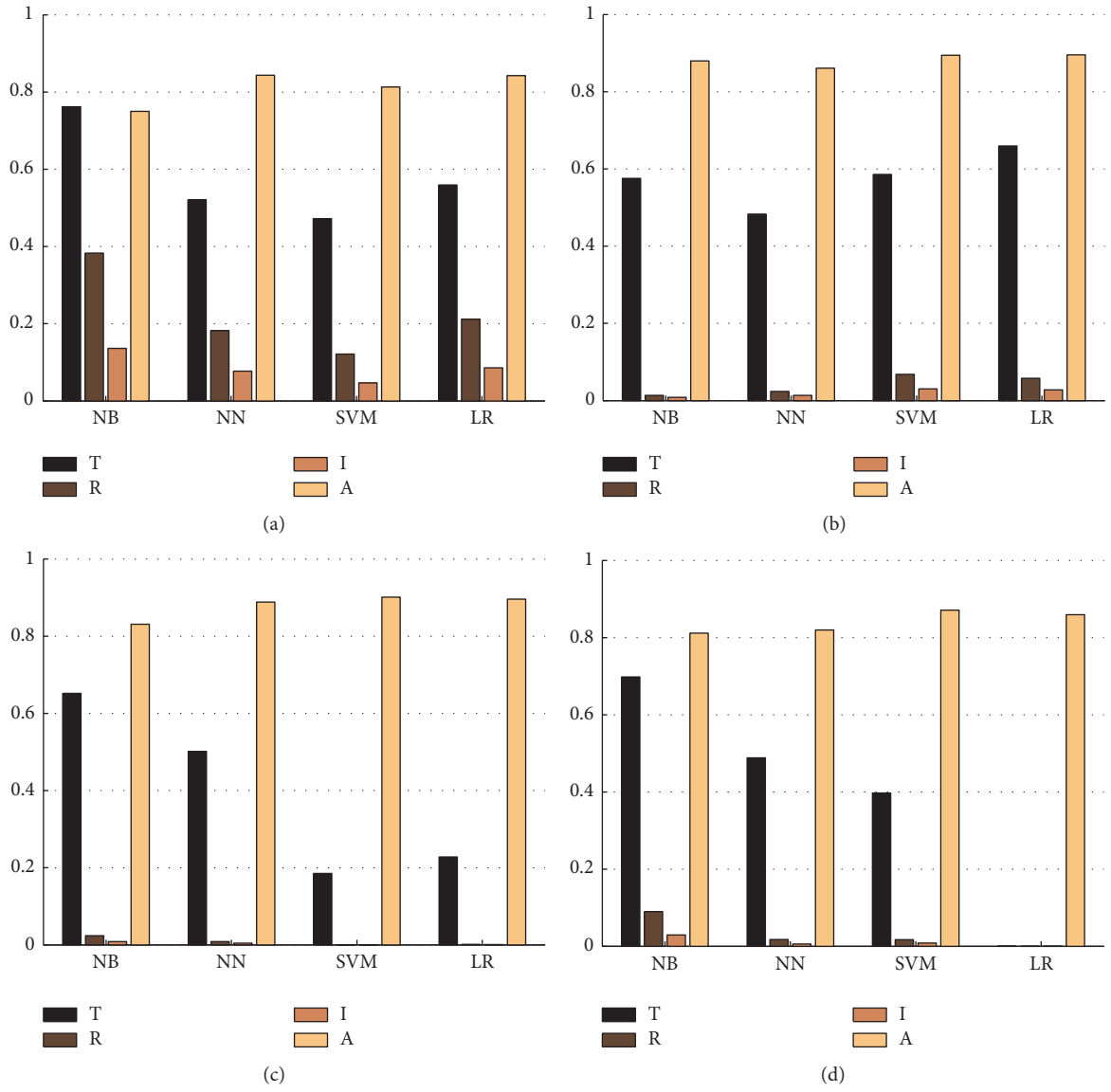

Figure 11: Rejecting samples with low classification confidence. T: confidence threshold, R: ratio of rejected samples to all samples, I: accuracy improvement obtained by rejecting low confidence samples, and A: classification accuracy within the samples with confidence ≥ threshold. Data representations: data set #1, (a) visits, (b) places, and (c) cumulative samples; (d) data set #2, cumulative samples. Accuracies given as ratios (unitless).

5.5. Effect of Number of Classes. In the previous tests we combined the less frequent places labels, such as friend's home, transportation, and restaurant into one class, *Other*. In this section, we compare these 3-class results to the 10-class classification results that we obtain with our classifiers and features. We used the same MDC data defined in Section 3.1 but now keeping the original 10 classes. We computed the places and cumulative samples representations from the 10-class data.

The comparison results are summarized in Table 3. We can notice that the numbers of cases are smaller in 3-class problem and the decrease comes from the decreased number of cases in classes other than *Home* or *Work*. We chose BT classifier for 10-class problem as it seems to outperform our other classifiers when number of classes is larger and compare it to LR of 3-class problem as LR performed well with both places and cumulative samples (Figure 3). For 10-class places representation, we computed two solutions, one using all the 14 features and another where we used forward SFS to select the most important features.

From the classification results it can be seen that adding more classes does not significantly affect the accuracy of *Home* and *Work*: the accuracies of *Home* are in both cases 92% or slightly better and the accuracies of *Work* are around 88%. However, the 10-class classifiers do not classify well the other places. With all features included, the overall accuracy is 62.3% and there are 4 classes that are never correctly classified. By reducing the number of features with SFS or

Table 3: Comparison of 3-class and 10-class solutions.

Number of classes	10			3	
Data representation	Places	Places	Cum. Samples	Places	Cum. Samples
Number of cases	369	369	128137	295	108531
Home: number (percentage)	106 (28.7)	106 (28.7)	39250 (30.6)	106 (35.9)	39250 (36.2)
Work: number (percentage)	98 (26.6)	98 (26.6)	37602 (29.4)	98 (33.2)	37602 (34.6)
Other: number (percentage)	165 (44.7)	165 (44.7)	51285 (40.0)	91 (30.9)	31679 (29.2)
Features	All 14	3: [4 1 2]	All 9	All 14	All 9
Classifier	BT	BT	BT	LR	LR
Overall accuracy (%)	62.3	68.5	68.4	89.2	89.5
Class accuracies (%)					
1 (Home)	92.4	92.4	94.6	93.0	92.0
2	61.5	57.6	54.1		
3 (Work)	90.8	89.7	91.7	92.0	88.0
4	25.0	62.5	45.4		
5	0.0	23.0	0.0		
6	0.0	22.7	5.0		
7	11.1	16.6	27.4		
8	0.0	20.0	13.0		
9	15.7	31.5	21.6		
10	0.0	42.8	23.2		
(Other)	(26.0)	(40.0)	(31.2)	81.0	89.0

Table 4: Comparison of data and solutions.

Solution	Users	Percentage of cases			Labels	Features	Best classifier	Accuracy (%)		
		Home	Work	Other				Overall	Home	Work
[6]	80	25	30	45	10	2,769,200	GBT	75.1	N/A	N/A
[7]	80	25	30	45	10	54	(1)	65.8	87	85
[8]	80	25	30	45	10	1177	(2)	73.3	100	100
[9]	114	25	29	46	10	500	(3)	75.5	92	90
#1 places	114	29	26	45	10	3 (SFS)	BT	68.5	92	90
#1 cum. s.	114	31	29	40	10	9	BT	68.4	94	92
#1 visits	114	52	38	10	3	14	NN	76.7	83	86
#1 places	114	36	33	31	3	14	LR	89.2	93	92
#1 cum. s.	114	36	35	29	3	9	LR	89.5	92	88
#2 cum. s.	16	31	26	42	3	11	LR	85.9	81	83

(1) Multilevel 2-method (SMO and simple logistic), fusion with decision tree. (2) Ensemble of binary classifiers using 1NN and SVM. (3) Combination of multiclass random forests and one-versus-all random forest binary classifiers.

using cumulative samples, the ability to classify also the less frequent places increases as shown in the bottom row, where average classification rates are computed for the other classes. Due to this improvement, the overall accuracy increases more than 6% to 68.5% and 68.4%. However, these are significantly lower than the overall accuracies of 3-class problem.

Based on this comparison, it is clear that with this type of user data, it is beneficial to combine the less frequent classes in order to classify better the more frequent and important places. Although the classification rates of *Home* and *Work* are on the same level in both 3-class and 10-class problems, the lower overall accuracies with 10-class indicate that there are more false detection of *Home* and *Work*.

6. Discussion

Papers [6–8] also aim at semantic place prediction and use data derived from the same database as data set #1 in our work. However, there are significant differences between their work and ours. Papers [6–8] are all from participants of the dedicated track on semantic place prediction in the Mobile Data Challenge (MDC) by Nokia, described in [5] and in more detail and with MDC outcomes in [36]. The data and findings based on it are described in [9], which also describes one solution of semantic place prediction. Basic information on the data, methods, and results of [6–9] and our work are summarized in Table 4.

The participants of the track used a subset of full MDC data that included the data of 80 users with the highest-quality location traces while we used the data of all the 114 users that had labeled visits data, without knowledge of quality of the data. The data used in [6–9] was from visits that lasted at least 10 minutes while our data was from visits that lasted at least 20 minutes. Therefore, their data included more cases from classes other than *Home* or *Work* compared to our data representations based on MDC data (data set #1). The difference is significant in visits representation but the accumulation of data changes these ratios. In data set #2, where the data collection has been implemented differently, the percentage of label *Other* is higher than the percentage of the other labels.

The numbers of extracted features are also given in Table 4. We used only 9–14 features related to time and phone usage but not to the environment while the other works used also environment related features such as number of Bluetooth or WLAN devices heard by the phone. We tested feature selection on data set #2 in both forward and backward directions but the results shown in the table were obtained using all 11 features. The authors in [7, 9] used feature selection method similar to our sequential feature selection in forward direction while in [6] they used two methods, Weka's Relief and L1-regularized logistic regression for the task.

The main focus in [6] is in generating a large number of conditioned features and then selecting the best features. The classification results using logistic regression, SVM with different kernels, Gradient Boosted Trees (GBT), and random forests are reported. The authors of [6] have published an extension to paper [21].

To give the final result, [7, 8] both use fusion of several classifiers or classification methods. Reference [7] uses multilevel classification model where labels are grouped so that in a sequence of classifications tasks with lower number of labels the algorithm selects label groups in hierarchical manner and finally in the lowest level chooses between two labels. In the paper, several methods are used to train different types of classifier models for multilevel classification. Then collection of these models is used to classify the data, and their classification results are used as a new feature vector that is used to train the final classifier.

Combination of smart binary classifiers is used in [8], where the multiclass classification problem is divided into a set of 2-class classification problems of types one-versus-one labels or one-versus-two labels. In the ensemble of binary classifiers each classifier uses the best combination of features for the current task and the better method from 1NN (i.e., KNN with $k = 1$) and SVM with RBF kernels. Three different methods for combining the classification outputs of the binary classifiers are evaluated in the paper.

In [9] three classification methods were used: (a) multiclass random forests, (b) one-versus-all random forest for each label where the winner class was decided by combining one-versus-all votes, and (c) combination of these. The accuracy of the methods was evaluated using leave-one-user-out cross validation similarly as in our comparisons.

In our work we solved 3-class problem with labels *Home*, *Work*, and *Other* instead of 10-class problem in [6–9]. We also used fewer features and simpler classifier models; that is, similarly as in [6] we did not use collections of classifiers except in BT (10 trees) and SVM (3 binary classifiers). The simpler models are generally preferred in resource constrained mobile devices. We also studied the effect of averaging of features by testing different data representations that include different levels of averaging: in visits representation each visit is classified separately, in cumulative samples, the features evolve with time as more data become available, and finally in places representation all data collected from one user in one place is averaged. For comparison, we also applied our features and classifiers to 10-class problem.

As we consider the memory consumption of SVM in classification phase too demanding for resource constrained mobile devices, we do not report its results in Table 4 even if it shows the best result with some data representations. In these cases, the results of the second best classifier are shown.

Due to the problem simplification from 10-class problem to 3-class in our approach and data retrieved from MDC database using slightly different criteria, the performance figures of Table 4 cannot be directly compared. However, due to the simpler task and despite the simpler classifier models, with visits representation and NN, we obtained the overall accuracy 76.7%, which is in the same level as the overall accuracy reported in the other works. With data representations including averaging the accuracies improve to 85.9% and better. The classification accuracies of *Home* and *Work* with places and cumulative samples of data set #1 are on the same level as in [7, 9]. With places representation where the data instances describe only short periods of time, these accuracies are lower as they are also with data set #2. In the latter case, the number of instances with label *Other* is higher than the numbers with the other labels, and, for this reason, the label *Other* is also classified with better accuracy (91%) than the two other labels.

The comparison between 3-class and 10-class problems with our classifiers and features show that our models can detect *Home* and *Work* reliably in both problems. The fact that in our model the inference is based on visits that are at least 20 minutes in duration may also contribute to this, as the shorter visits probably have phone usage characteristics that are closer to the decision borders. However, in 10-class problem the decreased classification rate of the less frequent places decreases the overall accuracy. Improving classification accuracy of the other places in 10-class problem requires using features that are directly related to environment, using phone usage data that is less privacy-preserving, and using more complex classifiers.

It can also be argued that MDC data is a bit old. As the MDC data set is from the time of the first smart phones, it does not describe well all the modern ways to use a smart phone. Through the evolution of new technologies, smart phone usage has changed a lot [37]. Nowadays, due to the internet connections available in phones, the use of SMS has decreased and messaging is often performed through other applications such as WhatsApp. The social media and messaging apps have reduced the need for voice calls and the

voice calls can also be made over internet based connections. Watching videos and TV on smart phones has become common as well as using social media and social games. With smart phones, photos are taken and videos recorded and both are shared in social media. Also the link between place and phone usage through the availability of WiFi networks is changing: the operators of wireless communication networks have started to bring inexpensive data plans with unlimited mobile data available to consumers, which allows them to use data-hungry applications also on the move [38].

7. Conclusion

We have developed an inference system to assign semantic place label for user's whereabouts based on the phone usage. The semantic places we considered in this work were Home, Work, and Other places. Our test results indicate that data representations that include averaging, that is, the places and cumulative samples representations, give higher classification accuracies than the visits representation. The average accuracy obtained using the visits representation is exceeded by the cumulative samples representation after only 2–6 days of accumulation of the data. Based on our preliminary tests with data set #2, the relevance of the features seem to depend on the classifier. However, features *stay*, *nightStay*, *headSet*, and *idle* seem to be relevant for most of the classifiers. Our tests also indicate that the classification accuracy can be improved by using thresholding based on classification confidence. The improvement was larger if the data representation did not include averaging.

7.1. Future Work. The future developments of the semantic labeling of user location context could include verification of the models using a bigger data set: more users, different life styles and daily patterns, different work occupations, and data for longer periods of time. The bigger data set could be used to learn subclasses to the current ones. In the group *Other* subclasses such as shop, restaurant, cinema, gym, outdoor exercising, lodging, leisure, and errands could be found. *Work* could include different kinds of work-like activities, such as shift work, driving work, other traveling work, attending school or university, and remote working from home. Also the use of *Home* is different for different people; for example, the elderly stay mainly at home.

In this study, we used bagged trees as an improved version of decision trees. Bagging improves variance of classifier by averaging/majority selection of outcome from multiple fully grown trees on variants of training set. Random forest is an interesting alternative for future work. It builds a collection of decorrelated trees by randomizing also the feature collection in the trees that are averaged (see, e.g., [31]).

Disclosure

This work is an extension to our paper [20] in UPINLBS 2014.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was financially supported by Microsoft Corporation and by EU FP7 Marie Curie Initial Training Network MULTI-POS (Multi-Technology Positioning Professionals) under Grant no. 31652. The research in this paper used the MDC database made available by Idiap Research Institute, Switzerland, and owned by Nokia.

References

[1] B. Heggestuen, "Smarthphone and tablet penetration," *Business Insider*, 2013, http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10.

[2] I. Lunden, "6.1b smartphone users globally by, 2020, overtaking basic fixed phone subscriptions," *TechCrunch*, 2015, https://techcrunch.com/2015/06/02/6-1b-smartphone-users-globally-by-2020-overtaking-basic-fixed-phone-subscriptions/#.t50cru:JF5k.

[3] B. Rao and L. Minakakis, "Evolution of mobile location-based services," *Communications of the ACM*, vol. 46, no. 12, pp. 61–65, 2003.

[4] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila, "Towards rich mobile phone datasets: Lausanne data collection campaign," in *Proc. ACM Int. Conf. on Pervasive Services (ICPS)*, Berlin, Germany, 2010.

[5] J. K. Laurila, D. Gatica-Perez, I. Aad et al., "The mobile data challenge: Big data for mobile computing research," in *Proc. Mobile Data Challenge by Nokia Workshop, in Conjunction with International Conference on Pervasive Computing*, Newcastle, UK, June 2012.

[6] Y. Zhu, E. Zhong, Z. Lu, and Q. Yang, "Feature engineering for place category classification , in," in *Proceedings of the Proc. Mobile Data Challenge by Nokia Workshop*, Newcastle, UK, Newcastle, UK, June 2012.

[7] C.-M. Huang, J.-C. Ying, and V. S. Tseng, "Mining users behaviors and environments for semantic place prediction," in *Proceedings of the Proc. Mobile Data Challenge by Nokia Workshop*, Newcastle, UK, June 2012.

[8] R. Montoli, A. M. Us, J. M. Sotoca, R. Montoliú, and A. M. Usó, "Semantic place prediction by combining smart binary classifiers," in *Proceedings of the Proc. Mobile Data Challenge by Nokia Workshop*, Newcastle, UK, June 2012.

[9] T. M. T. Do and D. Gatica-Perez, "The places of our lives: Visiting patterns and automatic labeling from longitudinal smartphone data," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 638–648, 2014.

[10] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Handheld and Ubiquitous Computing: First International Symposium, HUC '99 Karlsruhe, Germany, September 27–29, 1999 Proceedings*, vol. 1707 of *Lecture Notes in Computer Science*, pp. 304–307, Springer, Berlin, Germany, 1999.

[11] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.

[12] O. A. Nykänen and A. Rivero Rodriguez, "Problems in context-aware semantic computing," *International Journal of Interactive Mobile Technologies*, vol. 8, no. 3, pp. 32–39, 2014.

[13] J. Kantola, M. Perttunen, T. Leppänen, J. Collin, and J. Riekki, "Context awareness for GPS-enabled phones," in *Proceedings of the Institute of Navigation - International Technical Meeting (ITM '10)*, pp. 287–294, 2010.

[14] L. Pei, R. Chen, J. Liu et al., "Motion recognition assisted indoor wireless navigation on a mobile phone," in *Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS '10)*, pp. 3366–3375, 2010.

[15] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen, "IODetector: A generic service for indoor outdoor detection," in *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys '12)*, pp. 113–126, 2012.

[16] A. Eronen, J. Leppänen, J. Collin, J. Parviainen, and J. Bojja, *Method and apparatus for determining environmental context utilizing features obtained by multiple radio receivers, patent Application US0053069*, 2013, http://www.google.com/patents/US20130053069.

[17] Android Developers, "Locale object," http://developer.android.com/reference/java/util/Locale.html.

[18] T. Do and D. Gatica-Perez, "By their apps you shall understand them," in *Proceedings of the the 9th International Conference*, pp. 1–10, Limassol, Cyprus, December 2010.

[19] A. Rahmati, C. Shepard, C. Tossell, L. Zhong, and P. Kortum, "Practical context awareness: measuring and utilizing the context dependency of mobile usage," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1932–1946, 2015.

[20] A. Rivero-Rodriguez, H. Leppäkoski, and R. Piché, "Semantic labeling of places based on phone usage features using supervised learning," in *Proceedings of the Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS '14)*, pp. 97–102, 2014.

[21] Y. Zhu, E. Zhong, Z. Lu, and Q. Yang, "Feature engineering for semantic place prediction," *Pervasive and Mobile Computing*, vol. 9, no. 6, pp. 772–783, 2013.

[22] K. Farrahi and D. Gatica-Perez, "A probabilistic approach to mining mobile phone data sequences," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 223–238, 2014.

[23] T.-B. Nguyen, T. Nguyen, W. Luo, S. Venkatesh, and D. Phung, "Unsupervised inference of significant locations from WiFi data for understanding human dynamics," in *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia (MUM '14)*, pp. 232–235, November 2014.

[24] E. S. Lohan and P. Figueiredo e Silva, "User traces analysis based on crowdsourced data," in *Proceedings of the 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1303–1308, Valencia, Spain, June 2017.

[25] E. Malmi, T. M. T. Do, and D. Gatica-Perez, "From foursquare to my square: Learning check-in behavior from multiple sources," in *Proceedings of the ICWSM*, Boston, MA, USA, 2013.

[26] T. M. T. Do, O. Dousse, M. Miettinen, and D. Gatica-Perez, "A probabilistic kernel method for human mobility prediction with smartphones," *Pervasive and Mobile Computing*, vol. 20, pp. 13–28, 2015.

[27] Microsoft, "Lumia sensorcore sdk 1.1 preview," https://msdn.microsoft.com/en-us/library/dn924551.aspx.

[28] "Machine Learning, Neural and Statistical Classification," D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, Eds., Ellis Horwood, Upper Saddle River, NJ, USA, 1994.

[29] J. Han and M. Kamber, "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 2000.

[30] S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*, Pearson Education Inc, 2003.

[31] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, New York, NY, USA, 2008.

[32] K. J. Cios, W. Pedrycz, and R. W. Swiniarski, *Data Mining Methods for Knowledge Discovery*, Springer US, Boston, MA, USA, 1998.

[33] S. Haykin, *Neural Networks and Learning Machines*, Perason Education, Inc, 2008.

[34] S. J. Delany, P. Cunningham, D. Doyle, and A. Zamolotskikh, "Generating Estimates of Classification Confidence for a Case-Based Spam Filter," in *Case-Based Reasoning Research and Development*, vol. 3620 of *Lecture Notes in Computer Science*, pp. 177–190, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[35] W. Cheetham, "Case-Based Reasoning with Confidence," in *Advances in Case-Based Reasoning*, vol. 1898 of *Lecture Notes in Computer Science*, pp. 15–25, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[36] J. K. Laurila, D. Gatica-Perez, I. Aad et al., "From big smartphone data to worldwide research: the Mobile Data Challenge," *Pervasive and Mobile Computing*, vol. 9, no. 6, pp. 752–771, 2013.

[37] Deloitte, *There's no place like phone*, Deloitte Global Mobile Consumer Survey, 2016, http://www.deloitte.co.uk/mobileuk/assets/pdf/Deloitte-Mobile-Consumer-2016-There-is-no-place-like-phone.pdf.

[38] Tefficient, "Unlimited pushes data usage to new heights," *Industry analysis*, 2016, http://tefficient.com/unlimited-pushes-data-usage-to-new-heights/.