Classification of movement data concerning user's activity recognition via mobile phones

Spiridoula Tragopoulou
Harokopio University of Athens
Department of Informatics and Telematics
Omirou 9
17778 Tavros, Greece
stragopoulou@gmail.com

Iraklis Varlamis
Harokopio University of Athens
Department of Informatics and Telematics
Omirou 9
Tavros, 17778 , Greece
varlamis@hua.gr

Magdalini Eirinaki
San Jose State University
Computer Engineering Department
One Washington Square
San Jose, CA, 95192, USA
magdalini.eirinaki@sjsu.edu

ABSTRACT

Smartphones are becoming a powerful platform for event recognition due to the number of sensors they are equipped with. This provides an opportunity to apply data mining techniques on movement data in order to recognize people's daily activities without changing their routine. In this paper, we present a methodology for collecting and analysing user activity information with a smartphone application. This information can be further exploited in various applications ranging from m-health (e.g. fitness application) and transportation (e.g. user driving habits detection) to m-commerce (e.g. shopping recommendations). In order to demonstrate this methodology, we have developed GPSTracker[1] a prototype application for Android phones, which collects position, speed, altitude and time information and performs real-time classification of user's movement. The processed information is collected in a private user folder, on a cloud storage service, and can be further processed in order to extract aggregate user habits, or in order to detect user activity over time and provide recommendations. We also provide a visualisation of user trajectories, as recorded and classified by GPSTracker application. We exploited all the possible sensor of the smartphone and employed additional geo-location data from public transportation services, we tested several movement classification algorithms and trajectory pattern analysis techniques in order to improve the performance of our activity recognition process.

Keywords

activity recognition, classification, trajectory patterns, significant places

[1]A prototype of *GPSTracker* for Android (versions 2.2 and newer) can be downloaded from `http://galaxy.hua.gr/~it20934/`.

1. INTRODUCTION

Mobile phones have become an integral part of everyday life in modern society. The number of smartphone users increases rapidly, modern smartphones are equipped with multiple features such as cameras, GPS and other sensors and the specialized mobile applications cover almost every user need. According to MIT Technology Review the collection and analysis of information from simple cellphones was among the 10 breakthrough technologies of 2013[2], since it can provide surprising insights into how people move about and behave. In this direction, we decided to develop a mobile application that takes full advantage of the collected sensor data and creates a personal activities' repository, which can provide valuable information for the user's weekly activities and habits, her movement or driving style, the places that she usually spends time in weekdays or weekends etc.

In this prototype implementation for the Android OS, emphasis was given on solving technical issues concerning sensor data collection, real-time data processing in the mobile environment and data aggregation, in a private set-up, where every user can access her own data. For this reason we employed machine learning algorithms with minimum resource requirements, which can be executed in a mobile phone. We also employed smart trajectory correction techniques, that take into account changes in user movement (e.g. a stop in the traffic lights) in order to improve classification performance. *GPSTracker* uses publicly available cloud services for the private data repository (Dropbox) and the visualization of user trajectories (Google maps), which are also easily accessible from the mobile device.

Another reason why this application is different from previous ones is that all the data mining happens on the mobile device. While many different scenarios for mobile data mining exist [16], we chose to use the mobile device for the entire process. This allows the application to scale-up, since real-time processing of sensor data is performed in the mobile phone and only the processed data are moved to the cloud. As a consequence, the application in its current stage requires only storage resources from the cloud. In addition to this, it is up to the user's decision to share this information with other users or to make them available to post-

[2]http://www.technologyreview.com/featuredstory/513721/big-data-from-cheap-phones/

processing applications in the future. This increases the sense of privacy and user control over personal data. Due to the special characteristics that are unique to the mobile environment (such as low-bandwidth, limited battery power, slower processors) [10], this is a challenging task that introduces additional constraints to the original problem. In this paper, we explore several algorithms that satisfy the aforementioned constraints and show that there exist some that can process the user generated data on a stand-alone mobile device while maintaining high accuracy in the prediction process.

While this application prototype can be used as a stand-alone application, our vision is that it will be enhanced to become a "personal assistant" in that it will push notifications to the user depending on their predicted location or activity. For example, if the system predicts that the user is heading to the grocery store, it can push notifications including the user's shopping list, whereas if it predicts that the user is outside jogging, it can automatically suggest a playlist. Moreover, by collectively processing the application's user data, whilst guaranteeing user anonymity, we can also develop collective intelligence applications towards the same direction.

2. RELATED WORK

While sensors and/or GPS data have been used to track and predict mobility for a while, the opportunities arising by the broad usage of mobile devices has made such applications much more accessible for everyone to use. Using a smartphone's GPS receiver, accelerometer and/or magnetometer, one can gather real-time information about the user's position, direction, velocity, and, after some analysis, movement. Because of these capabilities, there's a growing market of health and fitness applications that focus on assisting users to measure their activity. Examples include MyTracks[3], Nike+ Running[4], Runkeeper[5], and MapMyRun[6] to name a few. All these applications use the phone's GPS and accelerometer to allow the user track and share their route, pace, distance, and time. Other applications, such as Automatic[7] and Waze[8], use the mobile device as a GPS and try to predict the user's driving routes.

While the exact details of the applications are proprietary, we may safely assume that most of these applications use simple heuristics to calculate the user's statistics of each route. Firstly, Activity Recognition[9] class provided by Android SDK, classifing movement between vehicle, bicycle, walking, still and tilting, is not always available and usually returns *unknown status*. More elaborate efforts, yet not implemented as fully-fledged applications, have appeared in the literature. A line of research deals with identifying and/or grouping moving object trajectories using Machine Learning techniques. For instance, Tsai et al.[19] used data collected from GPS sensors on wildlife animals to identify and group animal migration trajectories, using variable length Markov Models. Sung et al. [15] employed K-Means in order to perform trajectory clustering of moving robots and then used hidden Markov Models to predict their movement. Lee et al. [7] also proposed TRACLUS, a trajectory clustering algorithm. Another interesting application was that proposed by Trasarti et al. [18], who used the car GPS mobility data to identify the car driver's routes and match users based on their individual behaviors for carpools. None of the above approaches uses a mobile phone to collect and analyze the mobility data.

Many researchers map the mobility data to a coarser level before analyzing them. Sohn et al. [12] use coarse-grained GSM data from mobile phones to recognize high-level properties of user mobility and their daily step count using similarity metrics. Azam et al. [1] map the collected data to higher-level locations as well, before applying Neural Networks and Decision Trees to detect the behavior of low entropy people such as elderly people and patients in early stages of dementia. Taniar and Goh [17] use an apriori-like algorithm to predict moving patterns of mobile users. Liao et al. [9] use Conditional Random Fields on GPS data to segment user's day into activities and recognize and label significant places. Most of the approaches that employ data collected from the user's mobile phone focus are health and fitness application-oriented. Berchtold et al. [2] apply fuzzy classification and Kaghyan and Sarukhanyan [5] use K-Nearest Neighborhood to identify the user's activity. All of the aforementioned approaches require the data mining to be performed on the server.

However, in order to perform data mining on the server, a huge amount of sensor data must be transferred from every mobile device to the server, and the server must be able to handle all this processing. Many recent works focus on the idea of pre-processing data on the mobile device, thus moving part of the processing load from the server to the devices and reducing the amount of data transferred or removing the need for a permanent internet connection. Whereas there are clear trade-offs with regards to the capacity and computational power a mobile device provides, it has been shown that, given a proper prediction model, such systems can perform equally well while allowing the user to maintain their privacy and not share their data. Stahl et al. [13] discuss the idea of mining stream data on mobile phones. However, in their approach, they need more than one devices to distribute the mining process. Lee and Cho [8] have developed a system using hierarchical hidden Markov Models that allows real-time activity recognition on the Android OS to predict action and activity. While it is implied, it is not clear if all processing happens on the mobile device. Brezmes et al. [4] developed a system that tracks a user's movements, falling again under the health and fitness category. Their system employs K-Nearest Neighbors and all the processing happens on the mobile device. These two approaches are similar to ours in that the mining happens on the mobile device. The objective and therefore the choice of algorithms however is different. Moreover, none of the aforementioned research approaches provides a full-fledged mobile application prototype, integrating third-party applications, like the one we propose in this work.

[3]http://www.google.com/mobile/mytracks/
[4]http://nikeplus.nike.com/plus/products/gps_app/
[5]http://runkeeper.com/
[6]http://www.mapmyrun.com/
[7]http://www.automatic.com/
[8]https://www.waze.com/
[9]http://developer.android.com/training/location/activity-recognition.html

```
@attribute Longtitude numeric
@attribute Latitude numeric
@attribute ActualSpeed numeric
@attribute SmoothedAverageSpeed numeric
@attribute Altitude numeric
@attribute Timestamp date 'yyyy-MM-dd HH:mm:ss'
@attribute TimeZoneOffSet numeric
@attribute DayOfWeek {Mon,Tue,Wed,Thu,Fri,Sat,Sun}
@attribute IsWorkingDay {yes,no}
@attribute MoveType {Walking,Running,Biking,Driving,Metro,Bus,Motionless}
@attribute IsGpsFixed {yes,no}

@data
38.754539,22.857591,0.289095,1.507826,0.282153,'2013-08-07 21:39:34',2,Wed,yes,Biking,yes
38.753537,22.864428,0.519384,1.507826,0.282153,'2013-08-07 21:42:49',2,Wed,yes,Biking,yes
38.757827,22.858519,0.160138,1.507826,0.282153,'2013-08-07 21:46:04',2,Wed,yes,Biking,yes
```

Figure 1: Data recorded by GPSTracker.

Table 1: Features

Basic	*Derived*
Longitude Latitude	Average speed Smoothed average speed Near Metro Station
Altitude	Altitude change
Timestamp	Time zone Day of the week
GPS Signal status	-

3. APPLICATION ARCHITECTURE

The aim of *GPSTracker* prototype is to record and detect the daily routine of the user. Its architecture (shown in Figure 3) is based on four components, which implement data recording, classification of movement, visualisation of user tracks and data storage in the cloud respectively.

Data recording is performed by the *Location Service* component, an android service that records the three position coordinates of the user (i.e. latitude, longitude and altitude) every 30 seconds, using the GPS sensor of the mobile device. The service processes the GPS signal between two consecutive instances and calculates the average movement speed (called *Actual Speed* in the following) and the change in altitude. In order to avoid abrupt changes in user movement, such as a stop in the traffic lights, and provide a smoother trajectory, we also compute the average movement speed between the last three consecutive recordings (called *Smoothed Average Speed* in the following). Smoothed Average Speed also helps distinguishing between moving by bus or by car, since it results to greater values in the case of car driving. The timestamp of each recoding is also processed and two derived features are calculated, the *Day of the week* and the *Time zone*. Both attributes are used to distinguish better between peak and off-peak hours, since the speed of movement, especially by car or by bus, is expected to be strongly related to the traffic, which is increased during peak hours. Moreover, this transformation will allow us to distinguish between daily and weekend habits, between going to work or spending the night out etc. Recorded data is stored as arff files on the mobile device as shown in Figure 1.

Since the GPS signal is not always available, e.g. when moving inside a building or when moving underground by metro, *GPSTracker* also records the GPS signal status. This feature can be used in combination with longitude and latitude information in order to detect movement with metro. The set of all basic and derived features employed by *GPSTracher* is listed in Table 1.

Figure 2: The main screen of *GPSTracker*.

In order to record the GPS Signal status, we developed a custom solution, since Android does not have an appropriate service for this. More specifically, we have used information from the GpsStatus Listener, which is updated only when GPS status changes and the LocationListener, which receives the updated location information by the LocationManager Service every 1 second. On a GpsStatusChange event, we check the time that has passed from the last GPS signal, if the time is more than 5 seconds, we assume that the GPS signal is lost.

Recording sensor data during the whole day, needs that the application is always running, without user interference. However, the user must be able to switch to other applications or receive a call without interrupting data recording. For this reason the application has been implemented as a Service that runs in the background once started, in the current context battery consumption is not concerned. Data recording begins when the user presses the record button in the main screen (see Figure 2).

Movement classification is performed by the *Classification Activity* component. This android activity makes use of a lightweight implementation of the Weka API for Android mobile phones, and applies classification algorithms on movement data in order to recognize the user activity per movement instance. More specifically, every recorded instance (every 30 seconds) of the user activity is classified to one of the following movement types: Walking, Running, Biking, Driving, Metro, Bus and Motionless. A coarse grained classification of the aforementioned types to slow (i.e. walking, running, motionless) and fast motion (biking, driving, metro, bus) can be easily performed using the average speed, and maybe the altitude change information only. However, fine grained classification is harder and requires more information, which necessitates the need for more features. In an effort to add more useful features that will help us improve the classification performance, we added the boolean feature *Near Metro Station*, which examines the user distance from the closest metro station. The coordinates of all metro stations in the city of Athens and a distance calculation routine have been used for this purpose.

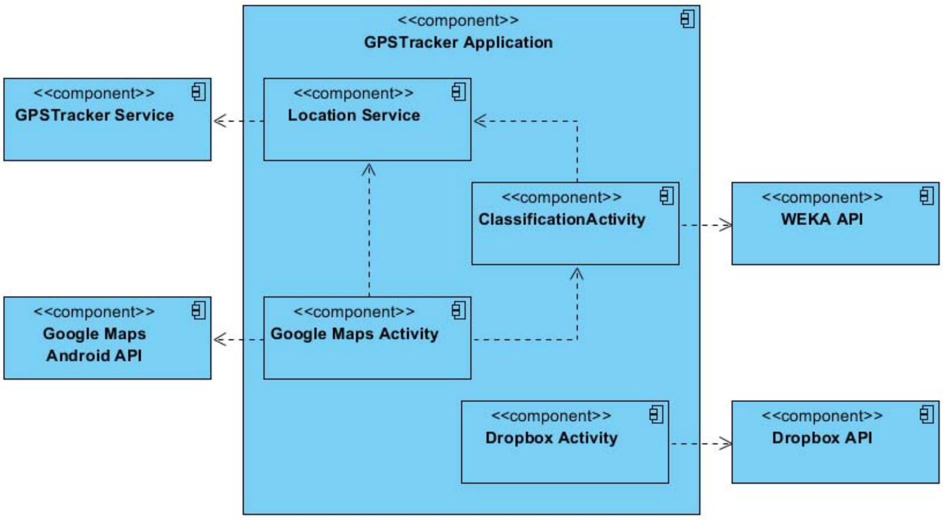

Figure 3: The architecture of *GPSTracker*.

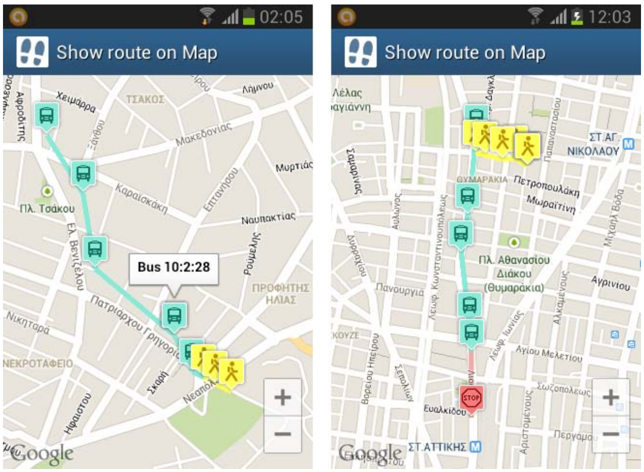

Figure 4: Real-time route rendering in *GPSTracker*.

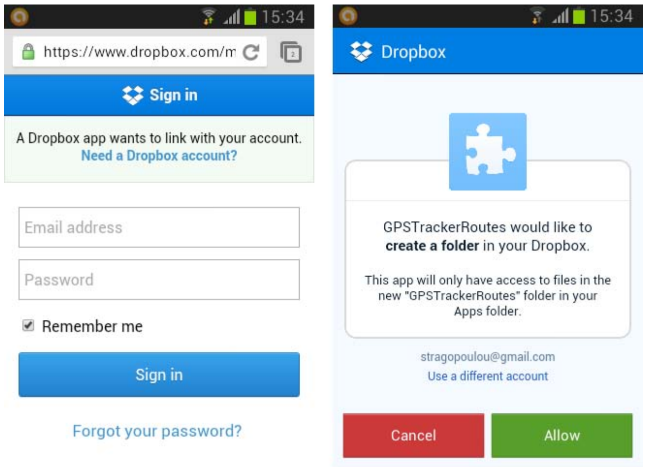

Figure 5: Dropbox account creation, via the Dropbox component.

Data recording and classification is performed as a background activity in the Android environment and the results are stored in the mobile device. At the same time, track visualization is performed at real-time on a map using Google Maps Activity component and the Google Maps API. Figure 4 presents snapshots of the application, which shows the route on the map when running in the foreground. In separate background activities, the application records information and classifies new instances.

Finally, the user can upload the files from the device to the cloud using the Dropbox API component, which runs in a separate Android activity. The API allows the user to create a Dropbox account (see Figure 5) and access it from the mobile device. Then the user is able to select and upload files to a central repository, thus emptying space in the mobile device (see Figure 6).

4. CLASSIFICATION EVALUATION

The classification of every recorded instance is performed using an implementation of the Random Forests algorithm [3] available by the Weka API. Random forests are an ensemble learning method for classification that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. The classifier was trained on a sample of 4518 instances distributed across the different movement types (see Table 2). The data have been collected using the *GPSTracker* training application depicted in Figure 7 and the trained model was embedded in *GPSTracker*.

The choice of Random Forests classifier was based on the availability of classification algorithms in the mobile version of Weka that we employed and on their performance in the evaluation set-up that we followed. Currently *GPSTracker*

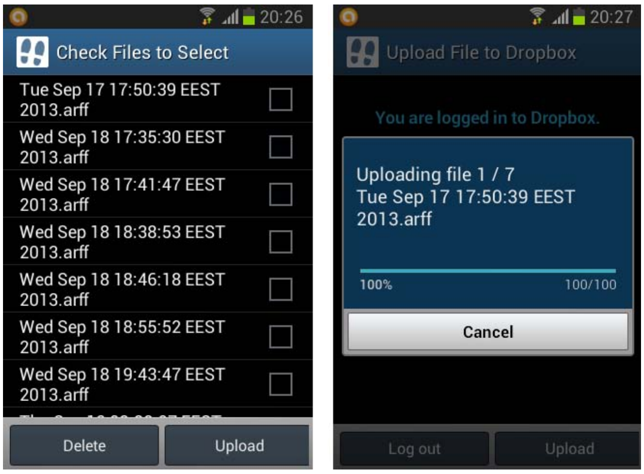

Figure 6: Uploading local files to Dropbox.

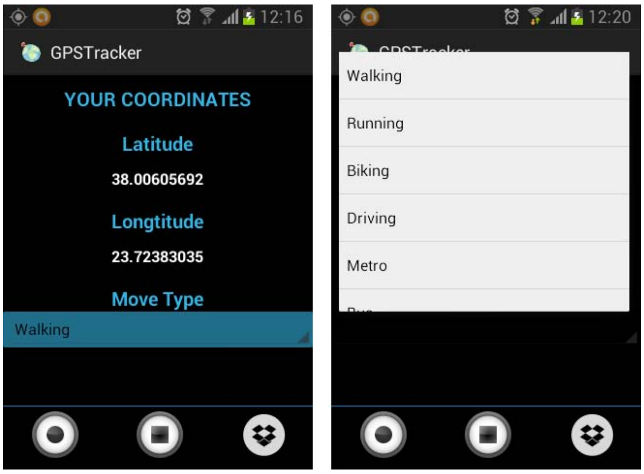

Figure 7: The *GPSTracker* training application.

uses a port of Weka 3[10] to the Android platform[11]. In the current implementation, we tested only tree-based classifiers, which are very fast in providing class predictions. More specifically we examined two C4.5 [11] implementations (J48 and REP Tree), the Logistic Model Tree (LMT) [6, 14] and Random Forests [3], with the default configuration parameters. The accuracy results when using 10-fold cross-validation on the entire data set are reported in Table 3.

As shown in Table 3, Random Forest algorithm has the best performance among the tested algorithms. In order to evaluate the generalization performance of our model and make sure that we avoid over-fitting, we draw the learning curves of our model using a holdout set, which we created by performing stratified sampling to the full dataset. The holdout (testing) dataset contains 20% of the full dataset instances and the remaining 80% was used for training. The error rate for the testing and training sets using an increasing number of training samples is depicted in Figure 8. From the learning curves it is obvious that the classification model generalizes well, since the error rate in the holdout set's instances decreases with the increase of training samples. In addition, the achieved error rate is quite low, which means that we can be quite confident for the movement type predictions of our model.

[10] http://www.cs.waikato.ac.nz/ml/weka/
[11] https://github.com/rjmarsan/Weka-for-Android

Table 2: Class distribution of the training samples

Attribute	Number of training samples
Walking	770
Running	177
Biking	343
Driving	650
Metro	1256
Bus	534
Motionless	788

Table 3: Accuracy in the training dataset using 10-fold cross validation

Algorithm	Accuracy
LMT	85.79
REP Tree	87.61
J48	90.73
Random Forests	92.81

In order to have a better understanding of what mislead our classifier, we performed an error analysis on the holdout samples. The confusion matrix depicted in Table 4 shows that the confusion was mainly among Motionless, Walking and Bus classes. This can be easily explained since the buses frequently stop, and similarly people occasionally stop during their walks (e.g. in traffic lights). We expect to solve this confusion when bus routes information will be added to the application and extra features concerning the user's position will be added to the classification model.

Table 4: Confusion matrix of the test data set.

a	b	c	d	e	f	g	<–classified as
135	2	0	2	2	2	11	a = Walking
1	32	0	0	0	0	1	b = Running
1	0	70	0	0	0	0	c = Biking
1	0	0	127	1	0	1	d = Driving
4	0	0	0	243	2	1	e = Metro
12	0	0	0	1	93	1	f = Bus
12	1	0	1	4	1	139	g = Motionless

5. FUTURE WORK

GPSTracker is a prototype application, which demonstrates the use of mobile phones for collecting user activity information. The next steps of our work comprise the extension of the set of features employed for motion classification, the long term analysis of user information and the collective post-processing of information from multiple users. The list of metro stations will be extended to include bus stations and all segments of the bus routes. This will allow us to create two more attributes, namely *Near Bus Station* and *On bus segment*. It is on our intention to add more geospatial information for public transportation stops and routes, parks,

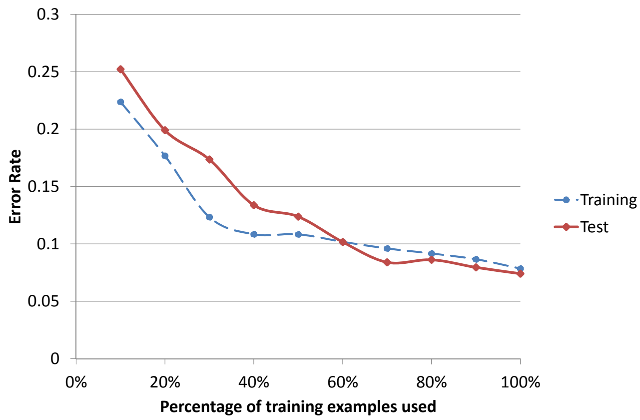

Figure 8: Learning curves for Random Forest algorithm.

malls and shopping areas and for this reason we will embed a geospatial extension of SQLite RDBMS. We currently design the analytical processing of multiple user trajectories, which must respect user anonymity whilst exploiting as much information as possible.

Another interesting direction we are working in, is the integration of this application with a personalized notification system. The application will be able to identify the user's path and type of transportation, and push related notifications (e.g. whether there has been an accident when the user is driving, or weather forecasts when the user is walking). This can become even more interesting when the social network element is added and the users can share paths and information with their friends.

6. REFERENCES

[1] M. A. Azam, J. Loo, S. K. A. Khan, M. Adeel, and W. Ejaz. Human behaviour analysis using data collected from mobile devices. *Int. J. on Advances in Life Sciences*, 4(1 and 2), 2012.

[2] M. Berchtold, M. Budde, D. Gordon, H. R. Schmidtke, and M. Beigl. Actiserv: Activity recognition service for mobile phones. In *ISWC*, pages 1–8. IEEE, 2010.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001.

[4] T. Brezmes, J.-L. Gorricho, and J. Cotrina. Activity recognition from accelerometer data on a mobile phone. In *Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, IWANN '09, pages 796–799, Berlin, Heidelberg, 2009. Springer-Verlag.

[5] S. Kaghyan and H. Sarukhanyan. Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axal accelerometer. *International Journal Information Models and Analyses*, 1, 2012.

[6] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. *Machine Learning*, 95(1-2):161–205, 2005.

[7] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 593–604. ACM, 2007.

[8] Y.-S. Lee and S.-B. Cho. Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer. In E. Corchado, M. Kurzynski, and M. Wozniak, editors, *Hybrid Artificial Intelligent Systems*, volume 6678 of *Lecture Notes in Computer Science*, pages 460–467. Springer Berlin Heidelberg, 2011.

[9] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26(1):119–134, Jan. 2007.

[10] S. Pittie, H. Kargupta, and B.-H. Park. Dependency detection in mobimine: A systems perspective. *Information SciencesŮInformatics and Computer Science: An International Journal - special issue: Knowledge discovery from distributed information sources*, 155(3-4):227–243, Oct. 2003.

[11] R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[12] T. Sohn, A. Varshavsky, A. LaMarca, M. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. Griswold, and E. Lara. Mobility detection using everyday gsm traces. In P. Dourish and A. Friday, editors, *UbiComp 2006: Ubiquitous Computing*, volume 4206 of *Lecture Notes in Computer Science*, pages 212–224. Springer Berlin Heidelberg, 2006.

[13] F. T. Stahl, M. M. Gaber, M. Bramer, and P. S. Yu. Pocket data mining: Towards collaborative data mining in mobile computing environments. In *IEEE International Conference on Tools with Artificial Intelligence (2)*, pages 323–330. IEEE Computer Society, 2010.

[14] M. Sumner, E. Frank, and M. Hall. Speeding up logistic model tree induction. In *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 675–683. Springer, 2005.

[15] C. Sung, D. Feldman, and D. Rus. Trajectory clustering for motion prediction. In *IROS*, pages 1547–1552. IEEE, 2012.

[16] D. Talia and P. Trunfio. *Mobile Data Mining on Small Devices through Web Services*, pages 264–276. John Wiley and Sons, Inc., 2010.

[17] D. Taniar and J. Goh. On mining movement pattern from mobile users. *International Journal of Distributed Sensor Networks*, 3(1):69–86, 2007.

[18] R. Trasarti, F. Pinelli, M. Nanni, and F. Giannotti. Mining mobility user profiles for car pooling. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1190–1198. ACM, 2011.

[19] H.-P. Tsai, D.-N. Yang, and M.-S. Chen. Mining group movement patterns for tracking moving objects efficiently. *IEEE Transactions on Knowledge and Data Engineering*, 23(2):266–281, 2011.