Semi-supervised learning: predicting activities in Android environment

Alexandre Lopes[1], **João Mendes-Moreira**[2,3], **João Gama**[3,4]

Abstract. Predicting activities from data gathered with sensors gained importance over the years with the objective of getting a better understanding of the human body. The purpose of this paper is to show that predicting activities on an Android phone is possible. We take into consideration different classifiers, their accuracy using different approaches (hierarchical and one step classification) and limitations of the mobile itself like battery and memory usage. A semi-supervised learning approach is taken in order to compare its results against supervised learning. The objective is to discover if the application can be adapted to the user providing a better solution for this problem. The activities predicted are the most usual in everyday life: walking, running, standing idle and sitting. An android prototype, embedding the software MOA, was developed to experimentally evaluate the ideas proposed here.

1 INTRODUCTION

Recognizing human activities with sensors next to the body has become more important over the years, aiming to create or improve systems in elder care support, health/fitness monitoring, and assisting those with cognitive disorders.

It is important to have systems that are practical for the user and that have the possibility to always be with them whilst not feeling strange or uncomfortable. Taking this into account we will attempt to use only one sensor instead of a, less practical but more accurate, system of distributed multi-sensors.

The new generation of smart phones has incorporated many powerful sensors, such as acceleration sensors (i.e. accelerometers), GPS sensors etc. They give the opportunity to create a system that can always be next to the user and work in real-time. In this work we will focus on the motion sensor of the cell phone, accelerometer, in order to predict the activity that the user is performing, as was attempted previously by Bao & Intille [1].

This problem will be treated as a classification problem using techniques of semi-supervised learning. This will be done in order to take advantage of existing examples (typically unlabeled) from the current user.

Knowledge discovery systems are constrained by three main limited resources: time, memory and sample size. In traditional applications of machine learning and statistics, sample size tends to be the dominant limitation. The problem of working with data streams is the arrival rate of the examples. When new examples arrive at a higher rate than they can be mined, the quantity of unused data grows without bounds as time progresses.

By building a new Smartphone application we attempt to solve problems consistent with previous undertakings, such as: accuracy, cost, performance among others. We explore matters like: (1) the impact of the app on the phone's battery lifetime; (2) how long should the interval to collect samples be in order to guarantee accurate classifications; (3) the time to create a model; and (4) the memory space needed.

All software used is open-source so the experiments can be continued and the application can be improved.

The aim of this work will be to create an application that adapt to each new user along time, learning his behavior and becoming more accurate.

2 RELATED WORK

Activity recognition is not new. Bao & Intille [1] created a system capable of recognizing twenty activities with bi-axial accelerometers positioned in five different locations of the user's person. This work led to an important discovery, which was possible to get accurate results predicting activities just using acceleration values gathered by a sensor placed on the thigh or dominant wrist. Despite this work uses twenty activities the most common activities used in other works [2,9,17] are walking, running, sitting, standing, up and downstairs.

Some research exists aiming to create a universal model that can be applied to any user. The idea is to use it in an Android application in order to measure the physical exercise of the user by predicting his activities [2]. This study uses three classification algorithms from WEKA (decision trees J48, logistic regression and multilayer neural networks) to induce models to predict user activities. Other studies, that also use the WEKA toolkit, implement common algorithms like Naïve Bayes, decision tables, K-nearest neighbors and SVM .

The common activities that research tries to predict are walking, running, sitting, standing, up and downstairs.

Gu et al. [3] tried to solve the activity recognition problem with techniques of semi-supervised learning using a large amount of unlabeled data, together with the labeled data, to build better classifiers. Because semi-supervised learning requires less human effort and gives higher accuracy, it is of great interest both in theory and in practice [4].

[1] Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, s/n 4200-465 Porto – Portugal, email: alexolopes89@gmail.com
[2] Departamento de Engenharia Informática, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, s/n 4200-465 Porto – Portugal, email: jmoreira@fe.up.pt
[3] LIAAD – INESC TEC, Rua de Ceuta, 118, 6º; 4050-190 Porto – Portugal
[4] Faculdade de Economia, Universidade do Porto, Rua Dr. Roberto Frias, s/n 4200-464 Porto - Portugal, email: jgama@fep.up.pt

One of the most important aspects of the research, in this field, is the classifiers' accuracy and the difficulty of label new instances. Both Masud et al. [15] and Guan et al. [16] use ensemble methods to increase accuracy in partially labeled data (semi-supervised problems). A common thing in all the works is how they try to find the more accurate model, testing multiple classifiers with the same data. Authors like Kwapisz et al. [2] showed, when trying to solve this classification problem using decision trees, that the most important attribute to differentiate the activities is the acceleration they induce on the accelerometer. Domingos et al. [11] showed that decision trees like C4.5 could be outperformed by Hoeffding trees, and demonstrated their importance when dealing with streams and limited memory space. The biggest problem of decision trees is that they assume that all training examples can be stored simultaneously in main memory, and are thus severely limited in the number of examples they can learn from. Still, regarding the accuracy, the problem can be solved in a hierarchical way. Hierarchical classification splits the initial problem into simpler sub-problems. The objective is to have a tree in the end where tests are done in each node. The classes contained in different nodes from the same level of the tree should be independent [5] so there is no possible uncertainty when choosing the path. It is expected to obtain more accurate classifiers by training them in the split data. For activity recognition, this can be done by classifying firstly whether the activity is motion or motionless and, in a second step, classifying it in lying, sitting, standing (if it was classified as motionless in the first step) or walking, gentle motion and posture translation (if it was classified as motion in the first step). These experiments came to the conclusion that rule-based reasoning can improve the overall accuracy proving the lustiness of this approach [6].

The main drawbacks of using such approaches in a mobile phone are the limited battery and memory. Experiments were carried out to determine how long the data samples provided by the cell accelerometer should be in order to obtain accurate classification. Some experiments were made and it was discovered that at least they need to be captured for 6s and the interval between them can be up to 10s [7]. These results are used in our experiments as described in section 4. Another thing that has impact on the cell phone, more specifically in its memory, is how the data is saved. Not all the data needs to be saved. Using sliding windows only the most recent data needs to be available [8]. The features of the raw accelerometer data that can be retrieved are the mean, the standard deviation, the energy and the correlation [9]. The usefulness of these features has already been demonstrated [1]. It allows saving both data and memory.

In terms of mobile applications, DiaTrace [10] is a system developed to aid in sport activities. The authors do not explain how they carry out the classification. However they guarantee 95% of accuracy if the mobile phone is used in the trousers front pocket. This is an example of how the market demands this type of applications.

3 METHODS

The tests were made on Naïve Bayes and Hoeffding Trees [11]. These two algorithms were chosen because some studies showed that Naïve Bayes can predict equally as well as decision trees (Langley, Iba, & Thomas 1992; Kononenko 1990; Pazzani 1996) and Hoeffding trees can learn in a very small constant time what is of major importance since we are dealing with streams in a mobile context.

The Naive Bayes algorithm is a classification algorithm based on Bayes rule and can often outperform more sophisticated classification methods. The Naive Bayes algorithm is based on conditional probabilities; it calculates a probability by counting the frequency of values and combinations of values in the historical data. Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. It assumes that the attributes X1…Xn are all conditionally independent of one another, given the target variable Y. The value of this assumption is that it simplifies dramatically the representation of P(X|Y), and the problem of estimating it from the training data [12]. An important advantage of this algorithm is the possibility to calculate the required probabilities in one pass over the training set. Additionally, it is able to obtain good classification performance even when trained in a small amount of data. We can conclude that this classifier can be trained on an efficient way, gathering the probabilities of each attribute

Hoeffding trees [13] operate by collecting, for each leaf node, sufficient statistics of the training instances each leaf contains. Periodically, these leaves are checked to compare the relative merits of each candidate attribute for splitting. The Hoeffding bound, or similar metric, is used to determine when a candidate is better than the others. At this point the leaf is split on the best attribute, allowing the tree to grow. Typically, information gain is used to rank the merits of the split candidates, although other metrics could be used. In the case of discrete attributes, it is sufficient to collect counts of attribute labels relative to class labels to compute the information gain afforded by a split. There are some variations of Hoeffding Trees, based on VFDT (Very Fast Decision Tree learner) which is a high-performance data mining system [11]. It is effective in taking advantage of massive numbers of examples by using a very small constant time per example. Since we are working with a mobile phone the biggest advantage is that Hoeffding trees do not store any examples (or parts thereof) in main memory, requiring only a space proportional to the size of the tree and the associated sufficient statistics [11].

The novelty of our work is the creation of the Android application that records data from the accelerometer. It uses a semi-supervised learning algorithm to process data with a model previously learned. This model is used to label the unlabeled data in real-time. This new labeled data can be used to train future models that fit over the user. In the semi supervised approach we defined a threshold of 70% (value that we assumed to be a good percentage of certainty for a classification) which means that we add to the training file the instances classified with 70% or more of certainty. We can also define the number of these new instances that we need to gather in order to create a new model. The older instances are deleted in order to maintain the size of the file.

4 AN ANDROID PROTOTYPE

We have implemented an Android application that records data from the accelerometer. We use: (1) sequence-based sliding windows [8] in order to save memory; and (2) the method of duty cycles [7] in order to save battery.

In sequence-based sliding windows an amount of data is defined. The file will have only the amount of data that the sequence-based sliding window allows. If new data is added it

replaces the oldest data in order to keep the size stipulated by the window.

In the duty cycles, 6s of data is needed in order to get enough data so an accurate classification can be achieved. To proceed with the classification we have 10s before retrieving new data. It means that the data from the accelerometer does not need to be fetched all time, saving battery with less operations of the app running. To sum up, we record data for 6s. Then, an instance is created with an average of the collected values. Finally, it is classified on the next 10s. This cycle is repeated along time.

4.1 EXPERIMENTAL SETUP

Before testing the application some decisions had to be made in order to have a controlled environment so we knew which result we were expecting for each test done.

The placement of the mobile was an important issue. Without having the option of placing sensors in different parts of the human body we have chosen the trousers' front pocket [14] to conduct all experiments. So there is recorded data with the mobile in a vertical and horizontal position inside the pocket.

To create the models, data from two persons was used. This data contained the average of the values recorded by an accelerometer for several hours doing, only, activities of walking, running, standing idle and sitting, being the waking activity the one with more recorded instances. In the total approximately 27 thousand instances were used.

The unlabeled data (files from approximately 16 thousand to 30 thousand instances) was not used to create the model. It belongs to the two people that contributed with data to create the model. There is, also, data from a third person that was not used for learning the models. It was used to evaluate the semi-supervised learning approach.

We needed to choose between timestamp and sequence-based sliding windows depending whether the window length is defined according to a predefined interval or a predefined amount of data. We have chosen sequence-based sliding windows because we wanted to keep the number of instances controlled and with a time interval that is impossible because the number of data elements in the window may vary over time.

A threshold of 70% probability is used to proceed with semi-supervised learning as explained in section 3. This allows creating new models by appending to previous data the recent labeled data when classified with 70% of certainty, at least.

4.2 EXPERIMENTS AND RESULTS

Previously, labeled data from three different persons was recorded. The data contained four activities: walking, running, sitting and standing idle. Using MOA, two different approaches were taken.

Firstly, models were induced using both Naïve Bayes and Hoeffding Tree. The classifiers were tested on unlabeled data from one person (Table 1).

Table 1. Classifiers' accuracy.

	Naïve Bayes	Hoeffding Tree
Accuracy	92.00%	94.78%

Secondly, a hierarchical approach with two levels was also carried out using the same classifiers. The hierarchical approach has two classifications: (1) The first one classifies the data into Dynamic or Static whether the activities involve motion or not, respectively (Table 2); (2) Then, in the second classification, a model was built on each category so we could proceed to the classification on Walking or Running on the Dynamic category, and Sitting or Standing Idle on the Static one (Table 3).

Table 2. Classifiers' accuracy in the first level of the hierarchical approach.

Dynamic vs. Static	Naïve Bayes	Hoeffding Tree
Accuracy	82.11 %	99.85%

Table 3. Classifiers' accuracy in the second classification of the hierarchical approach.

	Naïve Bayes	Hoeffding Tree
Running, walking	76.25%	99.05%
Sitting, standing idle	99.83%	99.93%

To test the effectiveness of the classification, unlabeled data of a person, which was not used for training the classifier, was used. Here are the results for the walking activity – Table 4.

Table 4. Accuracy for the walking activity using as test set data from a person without data on the training sets

	One-step classification	Hierarchical 1st classif.	Hierarchical 2nd classif.
Naïve Bayes	86,37%	90,17%	84,27%
Hoeffding Tree	67,65%	94,04%	88,09%

These results only show that Hoeffding Tree is better than Naïve Bayes for the walking activity on a hierarchical approach. However, Naïve Bayes gives better results on the one-step approach (Table 4). Further tests were needed for the remaining activities. Additionally, a semi-supervised approach was also used, besides the supervised one described above, in order to evaluate the usefulness of using unlabeled data from the user that is being tested.

In order to adapt the model to the normal user of the cell phone a threshold of 70% was created, as described in section 3. This meant that data labeled with at least 70% of certainty would be recorded on the training file of the classifier, so a new model, more suitable to the user, could be generated. This approach is compared against the supervised approach (Figure 1). It is easier to check the better accuracy when using the semi-supervised approach.

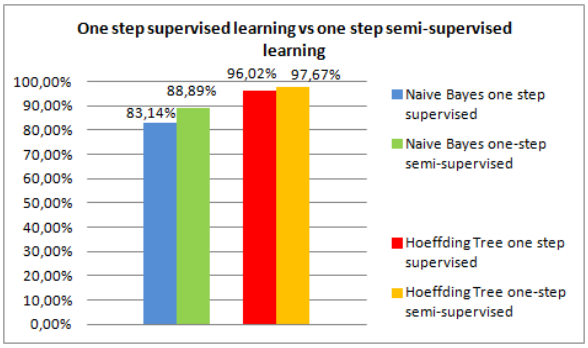

Figure 1 Accuracy of one step classification using both supervised and semi-supervised learning.

After doing the hierarchical classification (Figure 2 and 3) the labeled data was checked by visual inspection and it was easy to

observe that Hoeffding Tree tend to label data on the first classification as Dynamic (probably because the dataset is unbalanced and the Dynamic class is the majority one: there are about 15000 Dynamic instances and about 8000 Static ones). Naïve Bayes seems more balanced when labeling new data in the first classification of the hierarchical approach.

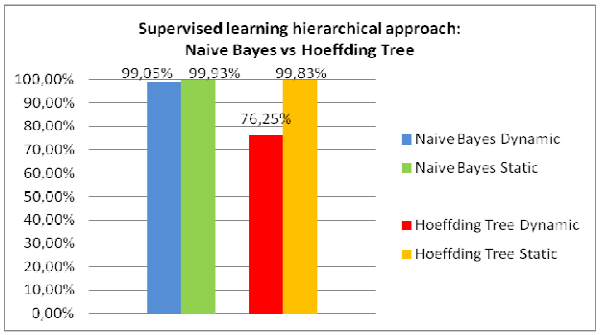

Figure 2. Classifiers' accuracy on final step of hierarchical classification with a supervised learning approach.

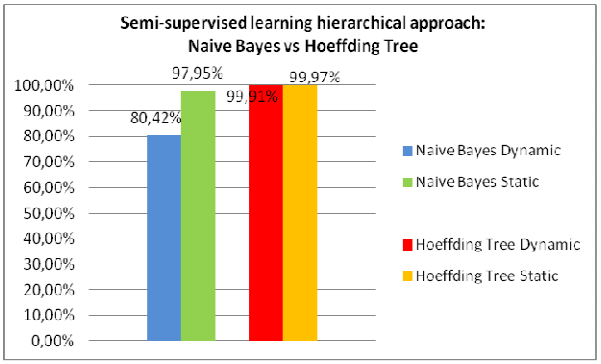

Figure 3. Classifiers' accuracy on final step of hierarchical classification with a semi-supervised learning approach.

At last we tested how using the two classifiers together would affect the classification (Figure 4).

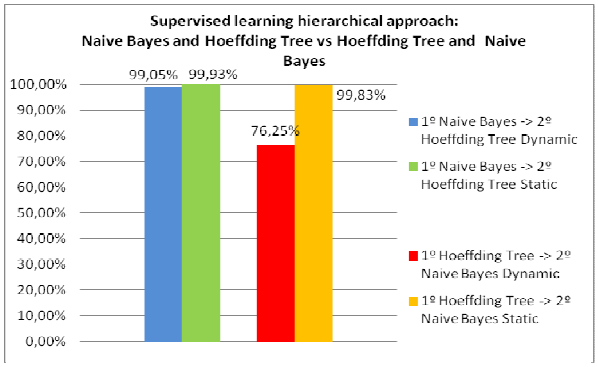

Figure 4. Classifiers' accuracy on final step of hierarchical classification with a supervised learning approach, using different classifiers for each step.

The balance characteristic of Naïve Bayes mentioned before can be verified in Figure 5, giving better results when used in the first classification. The tendency of Hoeffding Trees to classify, in the first step, the data as a Dynamic movement has influence on the second classification where Naïve Bayes has difficulties to label data because it gets lots of Static labeled data as Dynamic data from the first step. Overall better accuracy is achieved when using the Naïve Bayes classifier on the first classification (Dynamic or Static movement) and Hoeffding Tree on the second classification.

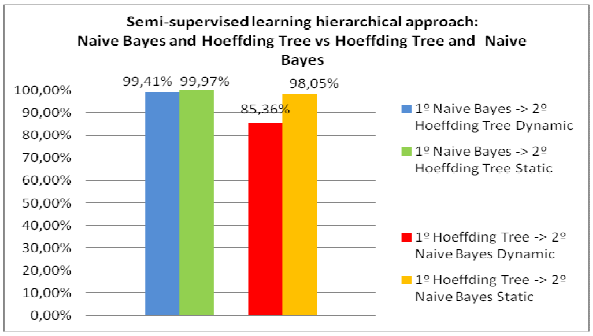

Figure 5. Classifiers' accuracy on final step of hierarchical classification with a semi-supervised learning approach, using different classifiers for each step.

The application had also concerns about both the battery and the memory usage. In order to test the battery usage, a stress situation where the app did both the hierarchical classification and the one step classification was created. In order to do it two models were created using the data of about 23.000 lines of labeled data, and doing the classification of 10 unlabeled instances. This experiment told us that the battery usage needs a maximum of 600.0mW for the CPU and between 500mW and 600mW for the LCD, which gives a total between 1100 and 1200mW on hierarchical classification. The one step classification only creates one model. The battery usage needs a maximum of 526mW for the CPU, the LCD needs the same power as the hierarchical approach, of course. Running the application five times, in a row, we got an energy usage of 120.8J for the CPU in hierarchical classification. However in one step classification we get a total of 110.3J.

Creating models and classifying about 10 instances took almost 60s which is a good time since we have only to classify 1 instance every 16s.

In terms of memory, the prototype is about 3Mb, and the files used for training the model having about 23000 lines are 1.466kB each. At most we will have the existence of three files for training (hierarchical approach). These files will grow because we defined a limit of 30000 instances for the training set (sequence-based window), which means that until we reach this limit none of the old training data will be erased and new data is added. When we reach the 30000 instances the sequence-based window will keep the size of the file. Whenever new labeled data from the user arrives (using the aforementioned 70% threshold) it will substitute the oldest data in order to have a semi-supervised learning approach.

The accuracy is not the only indicator of the classifiers' performance. Precision and recall are also important. The technique with higher accuracy might not be the one with the best balance between precision and recall. In our experiments we noticed that Hoeffding Trees have a better balance between precision and recall than Naïve Bayes.

5 CONCLUSIONS AND FUTURE WORK

The encouraging results of the experiments lead us to affirm that a step forward has been taken in the study of activity classification.

The most difficult activities to distinguish are walking and running because it is not clear where to draw the line between these two activities.

To achieve good results the techniques do not need to be too complex, like it was shown using Naïve Bayes. A fair conclusion after analyzing the figures is that hierarchical approach gives better results with Naïve Bayes doing the first classification and Hoeffding Tree dealing with the final one. With less complex techniques less power of the mobile is needed, leading to a minor impact on the classification performance. So, if Naïve Bayes does not decrease the accuracy it is better to use it in order to save memory and battery.

The battery usage confirms that the app can be used non-stop. It would be thrilling and of greater convenience to create a way that could swap classification techniques when the battery was low so it could be saved and the application did not have to stop. Changing from hierarchical classification to one step classification would have a maximum impact of 2% on the accuracy using Hoeffding tree as classifier.

The model used only has to be created when the application starts working. It is used for classifying until the app is shut down. It has only to classify one instance every 16s which is enough to do it, so the duty cycles work perfectly.

Regarding the memory usage a limit on the training files can be created, when this limit is reached the older data can be erased and new data added. This allows the adaptation of the application to new users as long as the application is being used by these new users.

The application can be improved by making possible to wear the mobile on other location, testing other classifiers or changing the way the data is processed.

New tests can be made using data from people with mobility constraints. Improving the app so it can adapt to this kind of people can be important if an accurate prediction can be made. Studies of patients with diseases that tend to degrade the ability to move can be accomplished to prevent, for example, falls or just to study how the movements change. This prevention can also be applied to elder people.

With this knowledge, people who practice sport can also benefit. For example, understanding how their body posture can be corrected in order to achieve better results.

This is just the beginning of an application that can be expanded in order to provide a better intimate experience between users and mobile phones.

ACKNOWLEDGEMENTS

This work is funded by the ERDF through the Programme COMPETE and by the Portuguese Government through FCT - Foundation for Science and Technology, project KDUS ref. PTDC/EIA-EIA/098355/2008.

REFERENCES

[1] L. Bao & S. S. Intille (2004). “Activity Recognition from User-Annotated Acceleration Data”, LNCS 3001, Springer, pp. 1-17.

[2] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity Recognition using Cell Phone Accelerometers,” *SIGKDD Explorations*, vol. 12, no. 2, pp. 74-82, 2010.

[3] Tao Gu, Zhanqing Wu, Xianping Tao, Hung Keng Pung, and Jian Lu. epSICAR: An Emerging Patterns based Approach to Sequential, Interleaved and Concurrent Activity Recognition. In Proc. of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (Percom '09), Galveston, Texas, March 9–13, 2009.

[4] X. Zhu, Semi-Supervised Learning Literature Survey, Tech. report, Computer Sciences, University of Wisconsin-Madison, USA, 2005.

[5] A. C. F. Coster, “Classification of basic daily movements using a triaxial accelerometer,” *Medical & Biological Engineering*, pp. 679-687, 2000.

[6] M. Schneider, M. Velten, and J. Haupert, “The ObjectRules Framework - Providing Ad Hoc Context-Dependent Assistance in Dynamic Environments,” *2010 Sixth International Conference on Intelligent Environments*, pp. 122-127, Jul. 2010.

[7] N. S. Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, “A Framework of Energy Efficient Mobile Sensing for automatic user state recognition”, pp. 179-191, 2009.

[8] B. Babcock and M. Datar, “Sampling from a moving window over streaming data,” *of the thirteenth annual ACM-SIAM*, 2002.

[9] N. Ravi, N. Dandekar, and P. Mysore, “Activity recognition from accelerometer data,” *Proceedings of the National*, pp. 1541-1546, 2005.

[10] G. Bieber, J. Voskamp, and B. Urban, “Activity Recognition for Everyday Life on Mobile Phones,” *Universal Access in HCI, Part II, HCII 2009, LNCS 5615*, pp. 289-296, 2009.

[11] P. Domingos & G. Hulten (2000). Mining high-speed data streams. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD’00*, 71-80. New York, New York, USA: ACM Press. doi:10.1145/347090.347107.

[12] T. Mitchell (1997). Machine Learning, McGraw Hill.

[13] Holmes, K. Richard, B. Pfahringer (2005). Tie-breaking in Hoeffding trees. In proceedings of the Second International Workshop on Knowledge Discovery from Data Streams, Porto, Portugal, 2005.

[14] S. Sprager and D. Zazula, “A cumulant-based method for gait identification using accelerometer data with principal component analysis and support vector machine,” *WSEAS Transactions on Signal Processing*, vol. 5, no. 11, pp. 369–378, 2009.

[15] M.M. Masud, J. Gao, L. Khan, J. Han and B. Thuraisingham, 2008. “A practical approach to classify evolving data streams: Training with limited amount of labeled data”. Proceedings of the 8th International Conference on Data Mining, December 15-19, 2008, Pisa, Italy, pp: 929-934.

[16] D. Guan , W. Yuan , Y. Lee , A. Gavrilov , S. Lee, “Activity Recognition Based on Semi-supervised Learning”, Proceedings of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, p.469-475, August 21-24, 2007.

[17] T. Brezmes, J. Gorricho, J. Cotrina, “ Activity Recognition from accelerometer Data on a Mobile Phone”, Lecture Notes in Computer Science, 2009, Volume 5518, Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, Pages 796-799