FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Smartphone Based Human Activity Prediction

Joana Raquel Cerqueira da Silva

Master in Bioengineering

Supervisor: Miguel Pimenta Monteiro (PhD)

23th July, 2013

Smartphone Based Human Activity Prediction

Joana Raquel Cerqueira da Silva

Master in Bioengineering

Approved in oral examination by the committee:

Chair: Artur Cardoso (PhD)
External Examiner: Jorge Alves da Silva (PhD)
Supervisor at FhP: Filipe Sousa (MSc)
Supervisor at FEUP: Miguel Pimenta Monteiro (PhD)

23th July, 2013

Abstract

Human physical activity monitoring has received an increasing interest by elders' caregivers, athletes, physicians, nutritionists, physiotherapists and even people who want to check the daily activity level.

Concerning applications for elderly, and taking into account the actual increasing of aging population and decreasing social and economic conditions for elderly daily care, telecare systems have emerging and have been considered as a solution for some of these problems.

In this project it will be described how to extract human postures, postural transitions and walking patterns from motion data recording with Smartphone built-in accelerometer, particularly a tri-axial sensor. The application of this system is to supervisor elderly or physical active people who are interesting in checking or improve their physical level. Methods to monitor activities of daily living, as standing, sitting, lying, walking and climbing stairs, were proposed based on a dataset composed of ten 60-70 years subjects, five male and five female, who carried a Smartphone placed longitudinally on their waist. The threshold-based approach implemented was capable of discriminating between static and dynamic activities. Static activities refer to situations when the user is static in a posture, as standing, sitting or lying; dynamic activities refer to activities that involve movement of the user, as walking, climbing stairs and transitions between postures (sit-to-stand movements). Within static activities, the angle between the user initial position and the orientation during each movement was used to differentiate the user postural orientation. The analysis of walking patterns were conducted in the frequency domain using Fast Fourier Transform and analysing the peak in the correspondent spectrum with higher amplitude, which corresponds to the step rate.

The problem of discriminating activities can be treated as a classification problem using techniques of machine learning. Using a public dataset provided by *SmartLab* during the ESANN competition, it was possible to study the more suitable metrics to extract from acceleration signals in order to train and test a classifier. Using a decision tree classifier, which was implemented with J48 algorithm from Weka, it was possible to achieve 86% accuracy and a 14% classifier error for the train and test datasets provided in the competition.

The results obtained suggest that accelerometer sensors could be used for accurate physical activity detection and strategies to implement the classification algorithms in Android environment should be implemented.

Resumo

Monitorizar a atividade física humana tem recebido um crescente interesse por parte de cuidadores de idosos, atletas, médicos, nutricionistas, fisioterapeutas e até por parte de pessoas interessadas em conhecer o seu nível de atividade diário.

No que diz respeito a aplicações para idosos, e tendo em conta o envelhecimento da população e a diminuição das condições sociais e económicas para cuidar dos idosos, os serviços de apoio à distância têm sido considerados como uma solução para estes problemas.

Neste projeto, vai ser descrito como extrair posturas humanas, transições entre posturas e padrões ao andar a partir de informação motora recolhida com o acelerómetro incorporado num *Smartphone*, particularmente um acelerómetro triaxial. A aplicação deste sistema baseia-se em supervisionar os idosos e pessoas ativas que pretendem controlar e melhorar a sua condição física. Os métodos para monitorizar atividades do dia-a-dia, como estar de pé, sentar, deitar, andar e subir escadas, foram propostos baseados num *dataset* recolhido com idosos entre os 60 e os 70 anos, cinco homens e cinco mulheres, que usaram o *Smartphone* colocado longitudinalmente na cinta. A abordagem baseada em limiares implementada foi capaz de diferenciar atividades estáticas e atividades dinâmicas. As atividades estáticas referem-se a situações em que o utilizador está estático em determinadas posturas, como estar de pé, sentado ou deitado; atividades dinâmicas referem-se a atividades que implicam o movimento do utilizador, como andar, subir e descer escadas e transições de pé-sentado-de pé. Para diferenciar posturas foi usado o ângulo entre a posição inicial do *Smartphone* e a orientação em cada movimento. A análise dos padrões de andar foi feita no domínio das frequências usando a Transformada de *Fourier* e analisando o pico no espectro da transformada com maior amplitude, que corresponde à cadência do andar.

O problema de diferenciar atividades pode ser tratado com um problema de classificação usando técnicas de *machine learning*. Usando um *dataset* público fornecido pelo *SmarLab* durante a competição *ESANN*, foi possível estudar as métricas mais adequadas a extrair dos sinais do acelerómetro, de modo a treinar e testar um classificador. Usando uma árvore de decisão, que foi implementada com o algoritmo J48 do *Weka*, foi possível obter uma precisão na classificação de 84% e um erro de 16% para os *datasets* de treino e de teste da competição.

Os resultados obtidos sugerem que os sensores do acelerómetro podem ser usados para determinar corretamente atividades humanas e estratégias para a implementação de algoritmos de classificação devem ser implementados em ambiente *Android.*

Acknowledgements

Prof. Doctor Miguel Pimenta Monteiro, FEUP supervisor, for his help and recommendations during the development of the project.

Eng. Filipe Sousa, Fraunhofer AICOS supervisor, for the insights and support in the different phases of the project.

Bruno Aguiar, Fraunhofer AICOS researcher, who promptly helped me solving problems and progresses with the project. It was important to share and learn with different points of view and more practical objectives of the project.

Tiago Rocha, Fraunhofer AICOS researcher, for his important support during the dataset collection, discussion of the results and help with the Android applications.

Jorge L. Reyes-Ortiz, from SmartLab, who promptly answered all my questions about the ESANN 2013 Competition Dataset.

To all the participants in the dataset collection, that kindly contributed to this project.

Joana Raquel Silva

Contents

List of Figures

List of Tables

Abbreviations

ADL	Activities of daily living
AR	Activity recognition
ARFF	Attribute-Relation File Format
DT	Dataset
FFT	Fast Fourier Transform
FEUP	*Faculdade de Engenharia da Universidade do Porto*
HAR	Human Activity Recognition
HMM	Hidden Markov Models
k-NN	k-Nearest Neighbours
SVM	Support Vector Machines
Weka	Waikato Environment for Knowledge Analysis.

Chapter 1

Introduction

Recognizing human activities with sensors next to the body has become an important research area, aiming to create or improve innovative applications providing activity monitoring. The ability to record and recognize individual daily activities is essential to determine the degree of functional performance and general level of activity of a person (Karantonis, Narayanan, Mathie, Lovell, & Celler, 2006).

These systems have real world applications in health care and fitness monitoring. Physical activity has positive effects on all body functions and studies proved that the risk of cardiovascular diseases is up to 50% lower on physical active people (Czabke, Marsch, & Lueth, 2011). With the progressive aging of population and the limited funding for public health care, more attention is paid to daily activity monitoring, improving the ability to assist the patients and help them care for themselves, reducing the conventional health care and moving towards the home telecare (Karantonis et al., 2006).

In health care field, long term analysis of human activity could be helpful in early detection of diseases (Czabke et al., 2011) or even to encourage people to improve their activity level. It could also be useful for physiotherapy, helping to understand if the recommended exercises are been correctly performed or even to assist those with cognitive disorders (Lopes, Mendes-Moreira, & Gama, 2012).

One of the most used approaches to monitor human activity is based on motion capture video systems that could also be associated with pressure plates in the ground. These methods are obtrusive, require massive devices and could only be used inside a laboratory environment, requiring a high set-up and processing time as well as memory space to record it (Czabke et al., 2011).

Motion sensors have become an interesting alternative to video systems, because of their miniaturization, low cost and capability to record motion signals within unobtrusive and wearable systems. Accelerometers and gyroscopes were also used in previously studies for daily activity monitoring (Czabke et al., 2011; Karantonis et al., 2006; Maurer, Smailagic, Siewiorek, &

Deisher, 2006), exercise information, such as energy expenditure (Lee, Khan, & Kim, 2011), and fall detection (Qiang et al., 2009).

Classifying motion information, collected with the sensors present in Smartphones, in activities labels is normally made with Machine Learning techniques that require the extraction of metric parameters of the motion data in order to train a classifier to predict for new data the activity it is associated with. The variability of people has reflexions in their mode to perform activities as simple as sitting or walking, and classifiers should also be general enough to deal with different users.

1.1 Background and Context

This dissertation is inserted in the data mining field, concerning human physical activity recording and classification, using motion sensors embedded in Smartphones. The project was developed in Fraunhofer AICOS Portugal. This research center focuses its activity in the area of assistive information and communication solutions to improve end-user experience and usability of applications (Fraunhofer, 2013b).

With the increasing aging of population and the decreasing of financial and social means to support and take care of elderly, people are more concerning with tele services, which are designed to help and improve the life of elders or even of people with some disabilities.

In this context, applications for human activity monitoring are emerging, not only for ones who want to attend their relatives and could not be present, but also for people who want to check the physical activity and improve it if necessary. Sports are also an important segment for physical monitoring applications, to follow the training profiles and change it according to their demands, or even to perform a gait analysis of their own and adapt rehabilitation exercises. For elite players this is an important issue, because the recovery should be as quick and accurate as possible. To monitor physical activity is important, not only for people with some disease, but also for healthy ones, who nowadays are, in general, more interested in maintaining an active life. In nutritional counselling, an exercise plan is sometimes defined with a set of activities to perform in a time period in order to lose weight. For each activity an average of the energy expenditure is calculated and the user is recommended to execute the activities necessary to achieve the defined plan[1].

1.2 Motivation and Objectives

The motivation behind this project is the implementation of a monitoring system that could help people tracking their physical activity and to help caregivers to assist the elders. The main objectives of this dissertation are to study the methods for activity monitoring capable of being implemented in a Smartphone, using a public dataset of activities of daily living (ADL) and a

[1] Consult Annex A for a more detailed explanation about activity monitoring and nutritional counselling.

Workbench of Machine Learning Technologies (Weka). In order to test those methods, it is necessary to collect a motion dataset, focus in ADLs performed by elders.

1.3 Project

This project aim to study the methods for monitoring human daily activities, using the motion sensors incorporated in a Smartphone. Focus was made in ADL as: standing, staying, lying, walking, walking up and walking down stairs. The movement of the user is captured by the accelerometers of the Smartphone placed in the user waist while he/she is performing the activities.

The project is also divided in two main areas of research: first it was studied the pre-processing and classifying techniques using a public dataset with the same activities of interest (standing, staying, lying, walking, walking up and walking down stairs), but with the advantage of having a large number of instances for each activity, providing better results for classifiers performance. This public dataset also had a predefined group of features that was treated as a basis of study for common features extracted for these types of activities. This study was made using Weka software.

Posteriorly, a dataset was collected at Fraunhofer, with older participants within an age range of 61-70 years. The referred activities were recorded with a Smartphone placed in the user waist in a supervised laboratory environment. Processing and evaluation of the data was done with MatLab® R2013a and Weka 3.6.9 software.

1.4 Overview of Dissertation

Apart from this Introduction (Chapter 1), this dissertation consists of four chapters. Chapter 2 is a review of the state of the art and explains some concepts related with activity monitoring and Machine Learning Technology. Some related previous studies are also presented. In Chapter 3, the activity recognition approach is described in detail. The results (Chapter 4) obtained with the proposed methodology are presented and discussed. In Chapter 5, the major achievements related with this dissertation are summarized and some improvements of the method are also pointed.

Chapter 2

Literature Review

This Chapter describes the state of the art related with the activity monitoring. Some previously studies in the area are also presented. The main technology concepts and resources used in the project are explained.

2.1 Activity Monitoring

The accurate monitoring of human activities has the potential to improve systems for health care, near-emergency early warning, fitness monitoring and assisted living (Lopes et al., 2012). These "motion-aware" systems can provide users with a wide range of add-value services: for example, by analysing the activities the user is performing during a period of time, it could be possible to determine trends of daily habits (Figo, Diniz, Ferreira, & Cardoso, 2010); recognition that an elderly person fell is important to trigger an emergency alert (Figure 1); it could also be useful for physiotherapy, helping to understand if the recommended exercises are been correctly performed or even to assist those with cognitive disorders (Lopes et al., 2012).

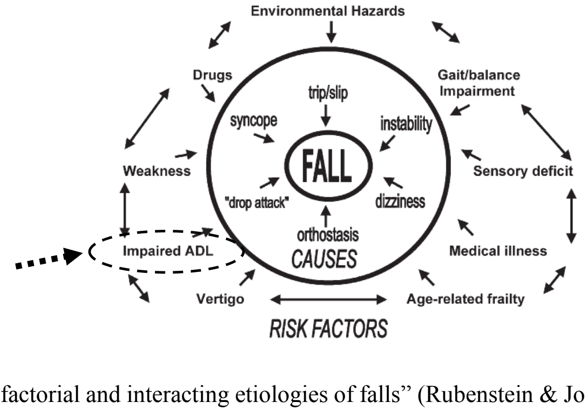


Figure 1: "The multifactorial and interacting etiologies of falls" (Rubenstein & Josephson, 2006), focus the context of this project related to the fall detection (dashed circle pointed with an arrow).

Activity concepts have been considered within Human-Computer Interaction (HCI) research in order to design better activity-based pervasive systems (Bao & Intille, 2004). The main goals of pervasive computing is ubiquity and unobtrusiveness, or to “fade into the background” supporting users while they perform daily activities (Wilde, 2011). Predicting activities is the objective of these pervasive systems, however the use of isolated actions for analysing real-life situations outside a laboratory is not successful, because actions are always situated into a context, and they are impossible to understand without that context (Kuuti, 1995). In the development of context-aware systems is important to recognize that an activity is the minimal meaningful context for understanding individual actions (Kuuti, 1995). Detecting the posture is not sufficient to differentiate between some activities; only if the context is captured, the activities could be accurately differentiated (Wilde, 2010).

The whole process for activity monitoring begins with gathering the raw data, in particular, motion data. Inertial sensors are an adequate solution to detect motion. These sensors respond to stimuli by generating signals that can be analysed and interpreted (Wilde, 2010). Usually, sensors are placed next to the body and should be comfortable for the user (Lopes et al., 2012). In section 2.2 some key aspects concerning sensors, as the type, number and their location, will be addressed in detail.

The new generation of Smartphones are equipped with a wide range of internal sensors, including accelerometers and gyroscopes, which can be used to monitor human daily activities. These devices are practical, small and unobtrusive, becoming an ideal platform for an activity recognition system. Other desirable features are the possibility to be wearable, work in real-time and be used for long-term monitoring (Lopes et al., 2012). These devices can acquire, process and obtain useful information from raw sensor data (Figo et al., 2010), but the key difficult of creating useful context-aware applications is to develop algorithms that can detect context from noisy and ambiguous sensor data (Bao & Intille, 2004).

Developing a Smartphone application has to take into account the limited resources of Smartphone as processing time, limited memory and sample rate and accelerometers are an ideal sensor because they require low processing power and energy consumption. In section 2.3, problems like the impact of the application on the phone’s battery lifetime, the correct acquisition sampling rate in order to guarantee accurate classifications and the limited memory space needed will be explored.

Some setup protocols for human activity data acquisition have been reported in the literature (Figo et al., 2010; Lopes et al., 2012), but other authors (Riboni & Bettini, 2010) also used a public dataset, like the COSAR dataset (from EveryWare Lab of the University of Milan) with annotated accelerometer data. A correctly annotated dataset is important for posterior system evaluation. Other studies (Bao & Intille, 2004) reported guidelines for the dataset recording, using different subjects, preferentially non-researchers or with no researcher supervision in order to not bias the data. Subjects were asked to perform a series of activities and to annotate the activities in the end (semi-naturalistic data collection) or to perform random sequences of pre-defined activities (specific activity data collection).

The activity prediction is normally treated as a classification problem, using techniques of machine learning based on probabilistic and statistical reasoning. Semi-supervised learning techniques require less human effort; give higher accuracy and enables the adaptation of the algorithm to each new user along the time (Lopes et al., 2012). "A supervised learning uses labelled data to train an algorithm, which then becomes able to classify unlabelled data" (Wilde, 2011). This technique has the following steps (Wilde, 2011): acquire sensor data, including labelled annotations; ascertain data features and its representation; aggregate data from multiple sources; divide data into training set and test set; train the algorithm on the train set; test the trained algorithm on the test set; apply the algorithm to new users. Section 2.4 will present the techniques normally used in each one of the steps for the classification algorithm implementation. Focus will be made to the classification features that could be used for Smartphone applications.

2.2 Sensors

Sensors can gather data that could be used for the detection of human activity. There are three main concerns regarding sensors: type, location and number. The majority of motion-aware systems have used inertial sensors, particularly accelerometers, to estimate the inclination of the body from the vertical and to determine the orientation and movement of the user (Wilde, 2010). Accelerometers use transducers for measuring linear acceleration (Figure 2). "Conceptually, an accelerometer behaves as a damped mass on a spring. When the accelerometer experiences acceleration, the mass is displaced to the point that the spring is able to accelerate the mass at the same rate as the casing. The displacement is then measured to give the acceleration" (Bird, Klein, & Loper, 2009).

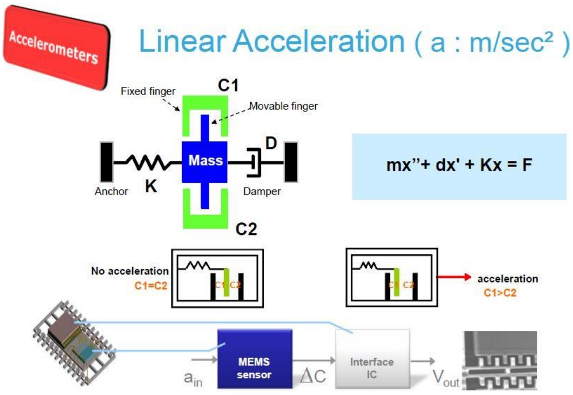


Figure 2: Linear acceleration sensors (Yu, 2013)

The signal obtained with accelerometers has two components, “a gravitational acceleration component (static) that provides information on the postural orientation of the subject, and a body acceleration component (dynamic) that provides information on the movement of the subject” (Mathie, 2003). A 3D accelerometer measures the acceleration along x (lateral), y (vertical) and z (longitudinal) axes relative to the screen of the phone as described in Figure 3. The acceleration is measured in SI (International System) units, m/s^2.

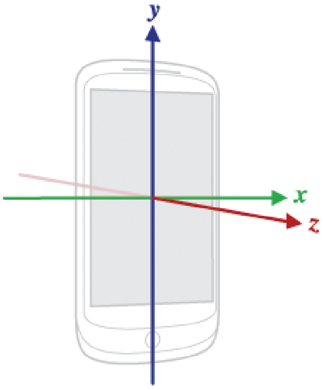
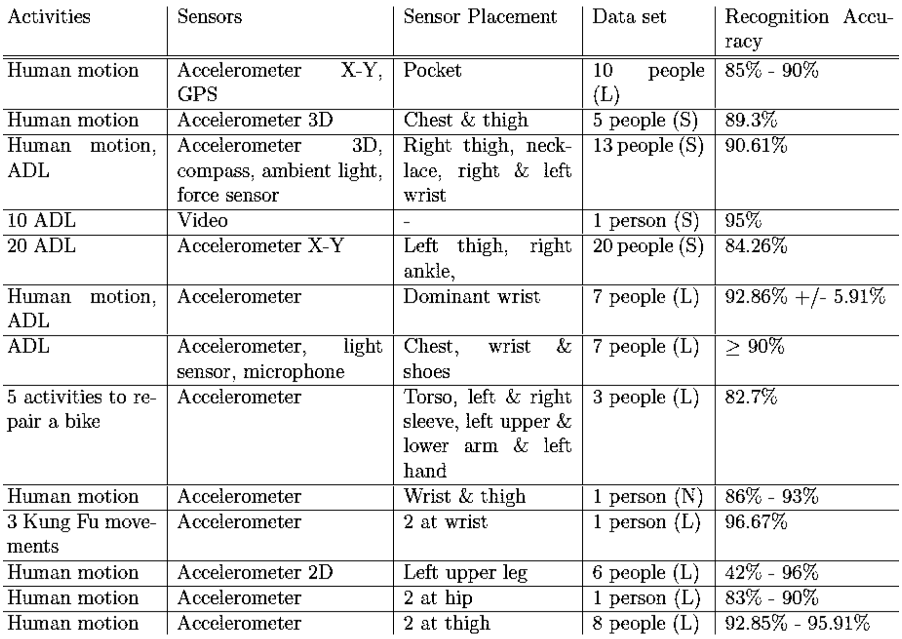


Figure 3: Smartphone axis orientation (CreatioSoft, 2012-2013).

In the literature, some investigations using activity counts and computer vision that support the potential for activity recognition using acceleration are also described (Bao & Intille, 2004). Many past works have demonstrated 85% to 95% recognition rates for ambulation, posture, and other activities using acceleration data (Bao, 2003). Some works[2] are summarized in Figure 4.



Activities	Sensors	Sensor Placement	Data set	Recognition Accuracy
Human motion	Accelerometer X-Y, GPS	Pocket	10 people (L)	85% - 90%
Human motion	Accelerometer 3D	Chest & thigh	5 people (S)	89.3%
Human motion, ADL	Accelerometer 3D, compass, ambient light, force sensor	Right thigh, necklace, right & left wrist	13 people (S)	90.61%
10 ADL	Video	-	1 person (S)	95%
20 ADL	Accelerometer X-Y	Left thigh, right ankle,	20 people (S)	84.26%
Human motion, ADL	Accelerometer	Dominant wrist	7 people (L)	92.86% +/- 5.91%
ADL	Accelerometer, light sensor, microphone	Chest, wrist & shoes	7 people (L)	$\geq$ 90%
5 activities to repair a bike	Accelerometer	Torso, left & right sleeve, left upper & lower arm & left hand	3 people (L)	82.7%
Human motion	Accelerometer	Wrist & thigh	1 person (N)	86% - 93%
3 Kung Fu movements	Accelerometer	2 at wrist	1 person (L)	96.67%
Human motion	Accelerometer 2D	Left upper leg	6 people (L)	42% - 96%
Human motion	Accelerometer	2 at hip	1 person (L)	83% - 90%
Human motion	Accelerometer	2 at thigh	8 people (L)	92.85% - 95.91%

Figure 4: Summary of past work on activity recognition using acceleration. “Data Set” column specifies whether data was collected under laboratory (L), naturalistic (N) or semi-naturalistic settings (Nadales, 2010).

[2] Consult Annex B for more detailed information about HAR based on accelerometers’ data.

According to Huynh (Huynh, 2008), researchers are more interested in using accelerometers because users consider accelerometers less intrusive than other sensors such as microphones or cameras. There are also many advantages that make this type of sensor useful for human activity recognition, as for example, their low cost and small size (make them adequate to embed in Smartphones) (Wilde, 2011).

Prior literature demonstrates that forms of locomotion such as walking, running, and climbing stairs and postures such as sitting, standing, and lying down can be recognized at 83% to 95% accuracy rates using hip, thigh, and ankle acceleration (Bao, 2003). However, the work of (Bao & Intille, 2004) suggests that thigh and dominant wrist are the better locations for accelerometer placement to detect ADL in naturalistic settings. Accelerometer data collected from dominant wrist is better for discriminating activities involving upper body movements and data from thigh accelerometer is useful for discriminate movements made with the lower limbs. Activity recognition systems should work with data from different placements, allowing the user to carry the device in the most convenient location for a given context (Wilde, 2011). When the sensors are in a single placement, it is useful to apply a multimodal information system to record the contextual cues of the environment (Wilde, 2010).

Concerning the number of sensors needed for an accurate recognition, (Bao & Intille, 2004) showed that using two sensors only affected 5% of the accuracy compared with a five-sensor system. Comparing bi-axial and tri-axial accelerometers, the extra cost does not enrich the data significantly (Bao & Intille, 2004). The maximum number of sensors reported in the literature was six uniaxial accelerometers (Bao, 2003).

Another important issue about sensors is that some activities as climbing stairs are usually indistinguishable using only accelerometer information, requiring complementary sensors (e.g. microphone and barometer) (Wilde, 2010).

The most common human activities addressed in the past works are walking, running, sitting, standing, going up and downstairs (Lopes et al., 2012). However, (Bao & Intille, 2004) reported a system capable of discriminate twenty activities, which lies outside the scope of this dissertation.

2.3 Smartphones

The new generation of Smartphones is being considered by users as an important personal device, together with an exponential availability. These devices have an increased potential for an adequate mean of gathering motion data, to use for building human activity prediction systems. The perception of their benefits are becoming commonplace, as users have become accustomed to their ubiquity (Wilde, 2011).

One of the most important applications of human activity prediction system is health care, particularly elderly health care. However, elderly people do not have the same skill in operating

Smartphones as young ones, and moreover do not carry these type of devices in daily living, becoming essential an adaptation phase prior to use of these systems, for elderly people to learn how to interact with the technology itself. Despite the fact that nowadays elderly are not used to interact with Smartphones, the future older users who have grown up with the technology will probably become an important market segment (Wilde, 2011).

Fraunhofer AICOS have already developed an application for Android Smartphones. The Smart Companion consists of a set of applications for the Android Smartphone that "were specially designed to meet older adults' needs. At the moment Smart Companion includes seven applications, namely: Home Smart Companion, Emergency, Calls, My Data, Music, Messages, Contacts" (Fraunhofer, 2013a).

Smartphones are equipped with a wide range of internal sensors (Table 1), including accelerometers and gyroscopes, which can be used to monitor human daily activities. These devices are practical, small and unobtrusive, becoming an ideal platform for a pervasive activity recognition system. Other desirable features are the possibility to be wearable, always next to the user, work in real-time and be used for long-term monitoring.

Table 1: Number of sensors in some recently launched Smartphones (Ristic, 2012).

Sensor Type	iPhone 4s	Galaxy S3	HTC One X	Droid 4	Lava Xolo 900
Accelerometer	●	●	●	●	●
Gyro	●	●	●	●	●
Dig. Compass	●	●	●	●	●
Proximity Sensor	●	●	●	●	●
Ambient Light Sensor	●	●		●	●
Barometer		●			

Android-based Smartphones have been chosen as the platform for this dissertation because the Android operating system is free, open-source, easy to program, and expected to become ordinary in people daily living. The low cost of the Smartphones that have accelerometers incorporated are also part of people nowadays routine.

The development of Smartphone applications for human activity prediction has some drawbacks as limited battery and memory and any new development on this area should address questions as: the impact of the application on phone's battery, sample rate in order to achieve an accurate classification, the time to create a model and its needed memory space.

Regarding the process of data collection, one could choose to collect the data with a Smartphone placed in predefined parts or just one part of the body (Section 2.2) and create an appropriate dataset, or to use public datasets, with annotated data. The former option is sometimes preferred because a well sized and annotated dataset is essential for evaluation of the classification algorithm, since an annotated dataset will be the ground truth wherewith it will be possible to check the results. To gather a sufficient complete dataset could not be feasible, because it is necessary to collect a great amount of data from several users doing diverse activities for a long period of time in order to create a robust and accurate classification algorithm that ideally could ultimately be adaptable to each new user. Even if one decides to create a dataset, voluntaries should be found to participate in the data collection, which is one of the biggest restraints of this process. (Bao, 2003) once said that "if individual variations in activity are significant, an activity recognition algorithm must be tested on data from many subjects to ensure it performs well for a range of individual patterns. Beyond the size of the data set, the quality of the data is also critical".

Some previous studies (Bao & Intille, 2004) reported guidelines for the dataset recording, using different subjects, preferentially non-researchers or with no researcher supervision in order to not bias the data. Subjects were asked to perform a series of activities and to annotate the activities in the end (semi-naturalistic data collection) or to perform random sequences of pre-defined activities (specific activity data collection).

Other studies (Riboni & Bettini, 2010; Wilde, 2011) used a public dataset COSAR dataset (from EveryWare Lab of the University of Milan) with annotated accelerometer data. This dataset have five hours of data collection from four participants that have performed ten physical activities[3]. The dataset also comprises data collected from two triaxial accelerometers (one on the hip and the other on the right wrist).

Another public dataset has been created as part of the ESANN 2012 Special Session in Human Activity and Motion Disorder Recognition (Anguita, Ghio, Oneto, Parra, & Reyes-Ortiz, 2012, 2013). The experiments have been carried out with a group of 30 volunteers within an age range of 19-48 years. Each person performed six activities[4] wearing a Smartphone (Samsung Galaxy S2) on the waist, placed longitudinally. Using its embedded accelerometer and gyroscope, they captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually (Anguita et al., 2012, 2013).

2.4 Technology

Once the dataset is ready, a classification algorithm needs to be implemented. As previously mentioned, human activity prediction is normally treated as a classification problem, using techniques of machine learning based on probabilistic and statistical reasoning. In 1959, Arthur Samuel defined machine learning as a "field of study that gives computers the ability to learn without being explicitly programmed". The basis of machine learning is exactly as Samuel has described, it builds a model and a classifier, capable of learning from unseen data. The model represents the data instances (normally each instance represents a data window with fixed size) and functions of these instances in the training step and ultimately the classifier could generalize

[3] The activities were: standing still, writing on a blackboard, brushing teeth, walking, walking downstairs, walking upstairs, climbing up, climbing down, running and riding a bicycle (Wilde, 2011).

[4] The activities were: walking, walking upstairs, walking downstairs, sitting, standing and lying (Anguita, Ghio et al. 2012).

for unseen data. Machine learning is a branch of computer science concerned with induction problems for which an underlying model for predictive or descriptive purposes has to be discovered, based on known properties learned from the training data[5].

Machine learning algorithms can be divided in several categories:

- **Semi-supervised learning** techniques enable the adaptation of the algorithm to each new user along the time (Lopes et al., 2012). Some example algorithms are Hidden Markov Models, naive Bayes networks, decision trees, K-Nearest Neighbours and Support Vector Machines (SVM).
- **Supervised learning** uses labelled data to train an algorithm, which then becomes able to classify unlabelled data (Wilde, 2011). The classification process could be implemented offline or online (Figure 5), according to the purpose. Offline is normally used for physical activity history recording, and online process is useful for inform the user or a caregiver in real time the current activity that is been performed.
- **Unsupervised learning** method tries to directly build recognition models from unlabelled data. Such an approach uses density estimation methods to discover groups of similar examples in order to create learning models (Wilde, 2010).
- **Ensemble learning** trains multiple learners to solve the same problem. Their generalization ability can be much better than that of a single learner (Zhou).

According to (Bao, 2003), “researchers must explore activity recognition systems using unsupervised data collection and training techniques for individual customization”.

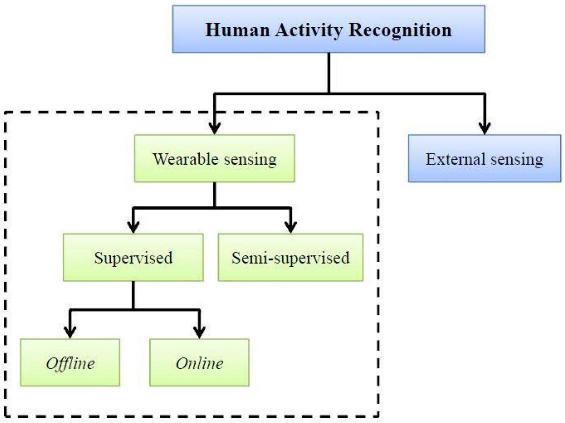


Figure 5: Taxonomy of HAR systems (Lara & Labrador, 2012).

[5] Sometimes data mining is confused with machine learning. Data mining is the extraction of patterns from data through statistics and artificial intelligence methods (Wilde, 2011).

Figure 6 summarizes the machine learning process that will also be implemented in this project.

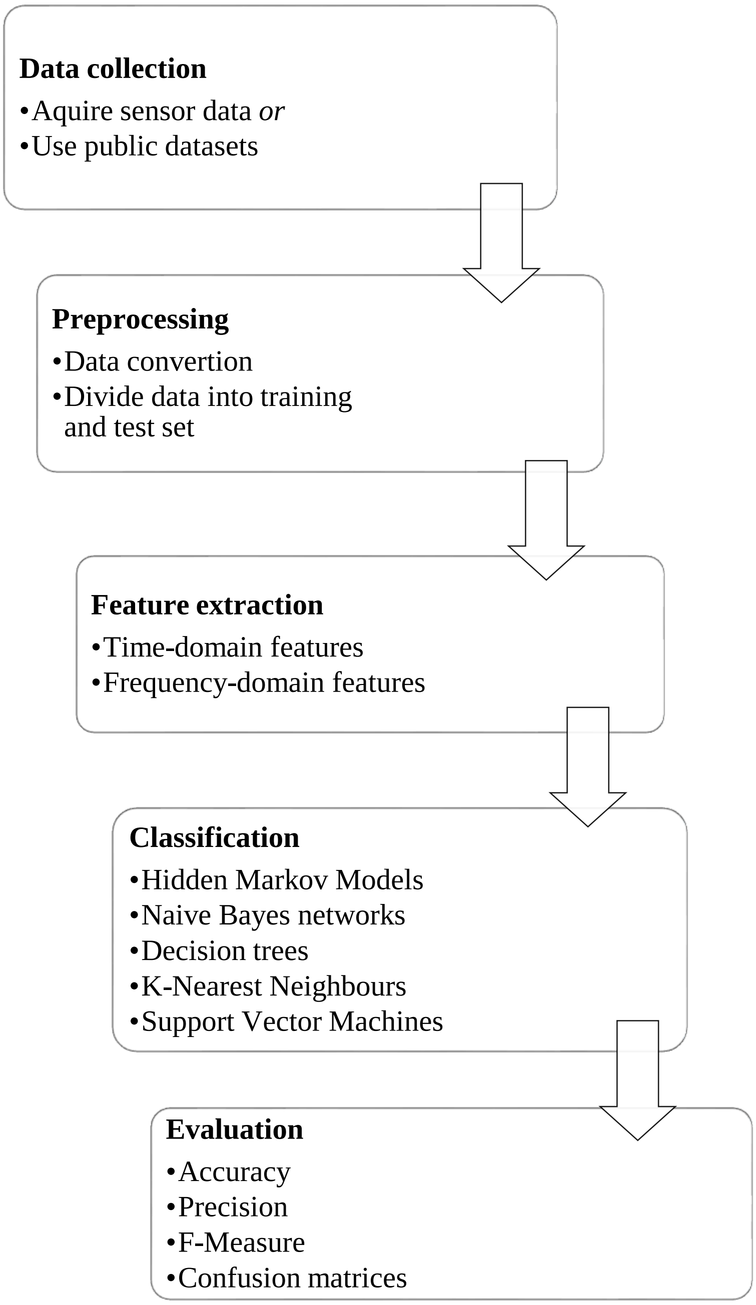


Figure 6: Schematic representation of a machine learning process. After data acquisition and preprocessing, the features need to be extracted in order to train the classification algorithm (the algorithms presented are some of many), that finally is evaluated, using confusion matrices.

Data collection has been covered over Section 2.2 and 2.3. The next four steps will be detailed in the following sub-sections. To implement these four steps, it was chosen to use the Weka workbench that contains more than 100 classification methods. Weka is a collection of state of the art machine learning algorithms for data mining (Wilde, 2010), developed in the University of Waikato, in New Zealand. This open-source collection includes tools for data pre-processing, classification, regression, clustering, association rules, and visualization, as well as utilities for algorithm evaluation (Wilde, 2010).

2.4.1 Pre-processing

The raw data usually needs to be pre-processed in order to be supplied to the Weka toolkit. Weka operates in ARFF format files, so every data file has to be converted into ARFF.

Accelerometer raw data needs to be divided in windows, sequentially, to be pre-processed. One should choose the window approach based on whether the recognition is intended to be done in real time (or “online") or not (Wilde, 2011). For online applications the window has to be defined in parallel with data collection, and for offline applications the window is defined prior to data collection (Wilde, 2010). The most common used approach is the sliding windows technique, where the signal is divided in equal windows with no gaps. However, this scheme suffers from the drawback that, as the window size is set arbitrarily, it might result in splitting the data in an inconvenient place, not capturing a “whole cycle" of the activity to be recognized (Wilde, 2011). This technique can be used with overlapping, normally 50% overlapping. (Bao & Intille, 2004) used a window size of 256 samples (corresponding to 5.12 seconds of data) and an overlap of 50%.

Raw data needs to be split into training and test set. The training set will be used to train the recognition algorithm and the test set will then be used to evaluate the algorithm after training. It is very important to not use test samples to train the classifier in order to not bias the result and ensure the classifier is evaluated in unseen data. Weka toolkit has some split data strategies, such as cross-validation, where a portion of the dataset is used for training and the rest for testing (avoiding a partitioning with classes overrepresented in the training set) (Wilde, 2011) and percentage split, where usually a third of the data is held out for testing, so the “train/test percentage split" is 66%.

2.4.2 Feature extraction

For each window, some features are extracted to characterize the signal. These features are then used as input for the recognition algorithms, to associate each window with an activity (Wilde, 2011). Time-domain, frequency-domain and symbolic strings domain features can be extracted from motion data. However, a combination of them is desirable.

Time-domain features are simple mathematical and statistical metrics used to extract basic signal information from raw data. It could also be calculated as data is being read. Usually, these features are simple to compute. Table 2 summarizes some of the time-domain features.

Table 2: Time-domain features. Respective formulas and applications (Figo et al., 2010).

Feature	Formula	Application
Mean		Data smoothing, axial calibration
Median	Separates the data into two halves	Discriminate different postures
Variance	Average of the squared differences from the mean	
Std Deviation	Square root of the variance	Signal stability
Min, Max, Range	Range is the difference between min and max values	Discriminate run and walk
Root mean square	$x_{RMS} = \sqrt{\frac{x_1^2+..+x_n^2}{n}}$ (1)	Distinguish walking patterns
Correlation,	$\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \, \sigma_y}$ (2)	Measure strength and direction of a linear relationship between two signals.
Cross-Correlation	$CrossCorr_{(x,y)} = {}^{n-1}_{d=1}\max\left(\frac{1}{n}\sum_{i=1}^{n} x_i \cdot y_{i-d}\right)$ (3)	Search for known pattern in a signal
Integration	Signal area under the data curve	Estimate speed and distance
Differences	Between signals in a pairwise arrangement	Compare signal strength in the 3 axis
Angular Velocity	Angle between acc-axis and gravity for the 3 axis	Determinate orientation, detect falls
Zero-Crossings	Points where signal passes half of signal range	Recognize step movements
Signal magnitude area	Sum of the area under the magnitude of each of the 3 axis acc-signal	Compute the energy expenditure in ADL
Signal vector magnitude	$SVM = \frac{1}{n}\sum_{i=1}^{n}\sqrt{x_i^2 + y_i^2 + z_i^2}$ (4)	Identify falls, monitor behaviour patterns
Differential SVM	$DSVM = \frac{1}{t}\left(\int_0^t (\sum\lvert SVM'\rvert)\, dt\right)$ (5)	Facilitate dynamic ADL classification

Frequency-domain techniques capture the respective nature of a sensor signal (Figo et al., 2010). In order to compute frequency-domain features, the sensor data window has to be transformed into frequency domain, using fast Fourier transform (FFT) (Wilde, 2011), which is a spectral representation of the signal. Another frequency-based representation, based on the decomposition of a set of orthonormal vectors, is the Wavelet Haar Transforms (Figo et al., 2010). In Table 3, some frequency-domain features are explained.

Table 3: Frequency-domain features. Their formula and applications (Figo et al., 2010).

Feature	Formula	Application
DC Component	First coefficient in spectral representation. Signal average	
Spectral Energy	Squared sum of its spectral coefficients normalized by the length of the sample window	Identify the way of transport
Information Entropy	Normalized information entropy of the discrete FFT coefficient magnitude excluding DC component	Differentiate between signals with same energy
Dominant frequency	Frequency value corresponding to the maximal spectral coefficient	Determine if a user is walking or running
Coefficients sum	Summation of a set of spectral coefficients	Recognition of some activities
Wavelet coefficients sum	Summation of all the coefficients of the Wavelet transform	Capture sudden signal changes. Detect falls

Another type of features are derived from **symbolic string-domain**, where accelerometer data is transformed into strings of discrete symbols (Figo et al., 2010), using a limited symbol alphabet to represent the signal. Symbolic aggregate approximation (SAX) uses piecewise aggregate approximation (PAA), which is a Gaussian equiprobable distribution function to map range values into string symbols (Figo et al., 2010). Table 4 summarize some of these features.

Table 4: Symbolic string-domain features. Their formula and applications (Figo et al., 2010).

Feature	Formula	Application
Euclidean-related distances	$ED\ (S,T) = \sqrt{\sum_{i=1}^{n}(\lvert s_i - t_i \rvert)^2}$ (6) (strings S and T)	Numeric distance between signal values that correspond to each string
Minimum distance	$MinDist =$ $\sqrt{\frac{n}{w}}\ \sqrt{\sum_{i=1}^{n} dist\ (s_i - t_i)^2}$ (7)	Quick discrimination of signals
Dynamic time warping	$DTW\ (S,T) = \min\left\{\frac{1}{K}\ \sqrt{\sum_{k=1}^{K} w_k}\right\}$ (8)	Measure similarity between two sequences

When applying these features computation on a Smartphone, one should be careful with their computational complexity, because of the Smartphones limited memory, processing capacity and battery lifetime. According to (Figo et al., 2010), almost all time-domain features are suitable for mobile devices, taken into account that correlation operations have a higher computational cost. With the exception of metrics based on Wavelet transforms, all others frequency-domain features are expensive in terms of computational cost. Symbolic string-domain metrics are very suitable for direct implementation on mobile devices because they require only integer arithmetic.

2.4.3 Classification

After extracting signal features, one should apply machine learning techniques in order to construct a classifier. It is possible to use Weka workbench to implement these recognition algorithms. Weka divides classifiers into lazy methods (k-NN), decision tree learners (C4.5), "Bayesian" methods (Naïve Bayes, Bayesian nets), function-based learners (SVM) and miscellaneous methods (meta-classifiers).

In the **k-NN** algorithm, the class assigned to a vector *x* is the class with the maximum number of votes coming from the *k* samples nearest to *x*. These classifiers are memory-based and do not require a model to fit (Campilho, 2009b). To find the closest samples the algorithm can use the Euclidean distance. The majority class of the k closest neighbours found is assigned to the test instance. It is a fast algorithm and the complexity is independent of the number of classes. In Weka, k-NN is implemented using IBk.

Decision trees use a tree in which for each attribute, one branch per each possible result of a test is generated. The algorithm stops when it finds a leaf, that represent a class (Wilde, 2011), and uses a divide-and-conquer approach. The Weka implementation of such an algorithm, such as C4.5, is J48. Some previously studies reported higher accuracy using these algorithms under user-specific training (Bao & Intille, 2004).

For **Bayesian classifiers**, when an object is erroneously classified can be quantified as a cost or a loss. And the expectation of the cost can be used as an optimization classifier (Campilho, 2009a). This algorithm represents the probability distributions of the training data.

SVM are linear classifiers that locate a separating hyperplane in the class space and classify points in that space. The objective is to find the maximum margin hyperplane separating two classes. The instances with minimum distance to the hyperplane are defined as support vectors. The computational cost increases as one wants to separate more classes. It defines the kernel function, which plays the role of the dot product in the class space. Weka offers several SVM algorithms, as a multilayer perceptron and SMO (implementation of a Supported Vector Machine with Sequential Minimal Optimization Algorithm).

(Wilde, 2010) reported that in spite of being the most accurate classifier, multilayer perceptron has high computational cost and longer training times, and k-nearest neighbours are more suitable for Smartphone applications. Decision trees are also suitable for Smartphone application given the simplicity and also their interpretable characteristic.

2.4.4 Evaluation

There are different ways to evaluate the performance of a recognition algorithm and most reported in the literature use confusion matrices (that is possible to determine with Weka toolkit), accuracy and precision.

Accuracy is the overall success rate, it is a measure of the global performance of the algorithm in what concerns correct decisions (Mendonça, 2009). **Precision** (IR precision in Weka workbench) is the fraction of detections that are relevant (Mendonça, 2009). **Recall "**refers how often the algorithm reports that an abnormality exists in the instances where it actually exists" (Mendonça, 2009). **F-measure** combines precision and recall by their harmonic mean (Mendonça, 2009) (Table 5).

Table 5: Performance metrics for classifications.

Metric	**Formula**
Accuracy	$(TP + TN)/(TP + TN + FP + FN)$ (9)
Precision (P)	$TP/(TP + FP)$ (10)
Recall (R)	$TP/(TP + FN)$ (11)
F-Measure	$2 \times (P \times R)/(P + R)$ (12)

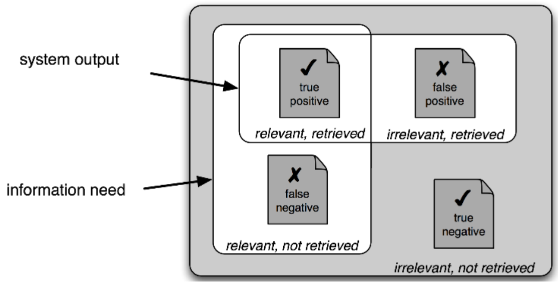
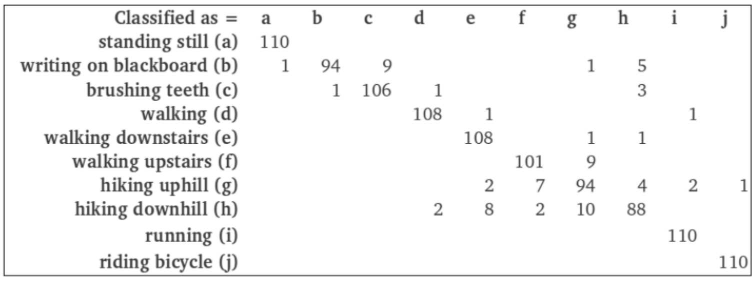


Figure 7: Confusion matrix, in a general view (Bird et al., 2009).

When dealing with a multi-class problem (each activity represents one class) the classification results can be represented in an *n-n* matrix of *n* classes, **confusion matrix** (Figure 7 and Figure 8). As seen in Figure 8, each number represents the number of instances belonging to class *i* classified as class *j* (Wilde, 2011). An ideal result is a diagonal matrix where *i=j* for each of the *n* classes.



Classified as =	a	b	c	d	e	f	g	h	i	j
standing still (a)	110									
writing on blackboard (b)	1	94	9				1	5		
brushing teeth (c)		1	106	1				3		
walking (d)				108	1				1	
walking downstairs (e)					108		1	1		
walking upstairs (f)						101	9			
hiking uphill (g)					2	7	94	4	2	1
hiking downhill (h)				2	8	2	10	88		
running (i)									110	
riding bicycle (j)										110

Figure 8: Confusion matrix from a previous study (Wilde, 2011).

2.5 Related Work

Current applications of HAR are explained in Table 6. Moves application for iPhone is an application that gives information about the kilometres and calories dispended during dynamic activities. eWatch is an activity recognition system implemented using the accelerometer and light sensor incorporated in a watch, applied for dynamic activities and also for sit and stand recognition, achieving an accuracy of 80%. MARS system was capable of real time physical activity recognition with activity confidence achieved in 30 seconds and display in pie charts. iLearn system, that uses iPhone's accelerometer and Nike+iPod sensors and recognition process has 97% accuracy. PAMSys motion sensor for Android is capable of distinguish posture orientations, postural transitions, detect falls and determinate gait parameters as step counter, using accelerometer sensors placed in a necklace.

Table 6: Current applications of HAR systems.

Application	Sensors	Activities	Solution	Accuracy
Moves for iPhone[6]	iPhone's sensors	Walk, run, cycling, step count	Unknown	Unknown

[6] (Karjalainen, 2013)

Context Awareness Activity Recognition (Intel)[7]	eWatch: accelerometer and light sensor.	Run, walk, up and down stairs, stand, sit	Decision tree	~80%
Mobile Activity Recognition System (MARS)[8]	Accelerometers	Run, walk, up and down stairs, still	Naïve Bayesian Classifier	Activity real time confidence
iLearn[9]	iPhone's accelerometer and Nike+iPod sensors	Running, walking, bicycling, sitting	Naïve Bayesian Network	97%
Physical Activity Monitoring (PAMSys)[10]	PAMSys motion sensor[11] (from BioSensics), Android	Posture, gait parameters, postural transition parameters	Unknown	Unknown

2.6 Summary and Conclusions

Concerning the purpose for which HAR system is designed, it is important to collect as much data of each activity as possible, but not only the quantity is important, the accuracy of the recording processing is also essential. Yang & Lianwen once said that "the recognition algorithms rely heavily on the dataset" (Ugulino, Velloso, Milidiú, & Fuks, 2012). The type, position and number of sensors used to collect data are also important and comparison of results between studies is only possible if the conditions are the same, which is not always possible to replicate.

The process required for human activity recognition is summarized in Figure 9 and basically it consists of five steps: collecting motion data for the activities of interest; pre-processing these data with filters, resampling in pre-defined windows for analysis and, in the case of using Weka for classification analysis, an additional step is needed to transform data into ARFF files; once data are upload to Weka workbench, it is necessary to select the more important features to input into the classifiers; after deciding which classifier is the more suitable for the purpose, evaluation metrics should to be computed, and the output of the classifier will give us for each instance of the data what is the correspondent activity.

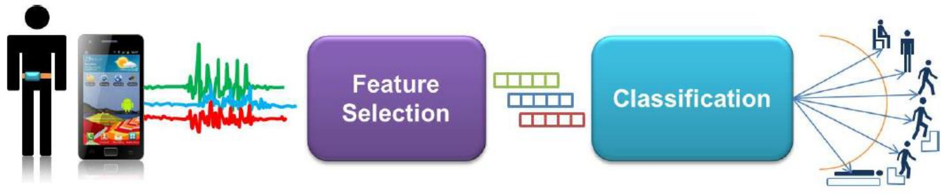


Figure 9: "Activity recognition process pipeline" (Anguita et al., 2012)

[7] (Maurer et al., 2006)
[8] (Gomes, Krishnaswamy, Gaber, Sousa, & Menasalvas, 2012)
[9] (Saponas, Lester, Froehlich, Fogarty, & Landay, 2008)
[10] (Research)
[11] tri-axial accelerometer, tri-axial gyroscope, tri-axial magnetometer

Chapter 3

Implementation

This chapter describes the methods used to evaluate accelerometer data, find patterns of activities and train the classifiers to predict for new data the correspondent activity.

Two approaches were employed: initially a public dataset, with enough instances both for training and test the classifier, was studied in to find suitable features for activity recognition and classifiers with the highest performance. In the scope of this chapter only the results of the reduced dataset, with the more important features, were presented, which is the basis for an activity recognition process using machine learning technology.

The second approach was based on a dataset collected at Fraunhofer AICOS, with elderly voluntaries. The process used to record the dataset is explained in detail. With these data, a threshold-based approach (classification tree) was proposed to discriminate the ADLs.

3.1 Machine Learning Technology

In the context of the present project, the activity recognition was performed by machine learning techniques. The purpose is to infer the activity of a person from sensor data streams. The input of the system is the data collected from a user performing the activities of interest, the methods used are the training of a classifier using extracted metrics from the input data and the output of the system is the prediction of the current activity for new unseen data.

3.1.1 Public Dataset

Context

The hypothesis of this dissertation is to test the possibility to implement a classification algorithm into a Smartphone for human activity prediction, taking into account battery and memory limitations. For this purpose, it was initially studied the techniques necessary for pre-processing and classifying accelerometer data into activities, using a competition dataset.

As initial stage of this project, it was studied the more suitable pre-processing and classification techniques using a public domain dataset for Human Activity Recognition, from the 21th European Symposium on Artificial Neural Networks, 2013, hereinafter designated competition dataset.

Characterization of the competition dataset

According to the authors of the competition dataset:

> "The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (walking, walking_upstairs, walking_downstairs, sitting, standing, lying) wearing a Smartphone (Samsung Galaxy S2) on the waist (Figure 10). Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data." (Anguita et al., 2013)

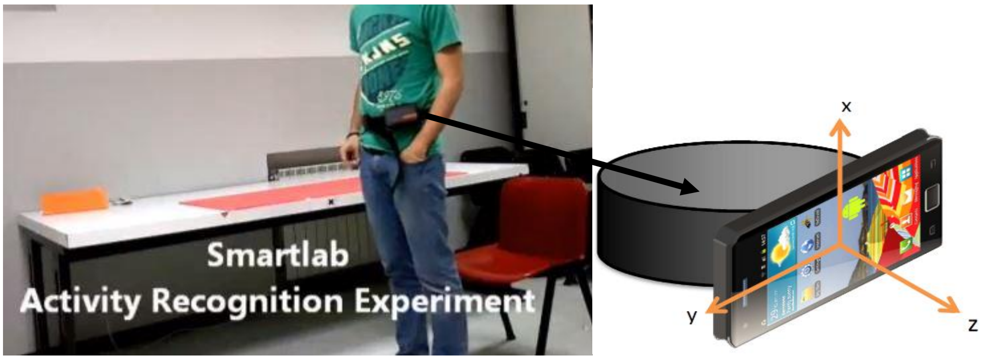


Figure 10: Set up for collecting data and Smartphone axis orientation (Anguita et al., 2013).

The authors provided the accelerometer and gyroscope signals for each dataset (train and test) as well as the correspondent activity label for each instance.

> "The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components; therefore a filter with 0.3 Hz cut-off frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain." (Anguita et al., 2013)

The features were normalized and bounded within [-1, 1]. The normalization performed was according to equation 13:

$$\boldsymbol{xNorm} = \frac{\boldsymbol{x - Min}}{\boldsymbol{Max - Min}} \times \boldsymbol{2} - \boldsymbol{1} \tag{13}$$

Where *xNorm* is the feature normalized, *x* is the feature and *Min* and *Max* are the minimum and maximum values, respectively, of feature *x*.

For a more accurate classification, the datasets should be balanced and all the six classes should have the same number of instances. The class distribution from the competition dataset is represented in Table 7.

Table 7: Number of instances for each class, for training and test sets.

Class	Training set	Test set
1- Walking	1226	496
2- Walking down stairs	1073	471
3- Walking up stairs	986	420
4- Sitting	1286	491
5- Standing	1374	532
6- Lying	1407	537
(Total)	7352	2947

The authors extracted 561 features[12] that correspond to the 17 features of Table 8, extracted over the signals of Table 9:

Table 8: Extracted features and their description (Anguita et al., 2013).

Feature	Description
mean()	Mean value
std()	Standard deviation
mad()	Median absolute deviation
max()	Largest value in array
min()	Smallest value in array
sma()	Signal magnitude area
energy()	Energy measure. Sum of the squares divided by the number of values.
iqr()	Interquartile range
entropy()	Signal entropy
arCoeff()	Autorregresion coefficients with Burg order equal to 4
correlation()	Correlation coefficient between two signals
maxInds()	Index of the frequency component with largest magnitude
meanFreq()	Weighted average of the frequency components to obtain a mean frequency
skewness()	Skewness of the frequency domain signal
kurtosis()	Kurtosis of the frequency domain signal
bandsEnergy()	Energy of a frequency interval within the 64 bins of the FFT of each window.
angle()	Angle between two vectors.

[12] Consult Annex C for a more detailed explanation about the 561 features extracted from this dataset.

Table 9: Signals derived from the accelerometer data, used in time and in frequency domain (Anguita et al., 2013).

Time domain	Frequency domain
tBodyAcc-XYZ	fBodyAcc-XYZ
tGravityAcc-XYZ	fBodyAccJerk-XYZ
tBodyAccJerk-XYZ	fBodyGyro-XYZ
tBodyGyro-XYZ	fBodyAccMag
tBodyGyroJerk-XYZ	fBodyAccJerkMag
tBodyAccMag	fBodyGyroMag
tGravityAccMag	fBodyGyroJerkMag
tBodyAccJerkMag	
tBodyGyroMag	
tBodyGyroJerkMag	

3.1.2 Weka Workbench

Concerning pre-processing techniques, there are two methods for *feature*[13] *selection*: filters and wrapper. The later creates all possible subsets from the feature vector and use classifiers on each subset, the objective is to select the subset with which the classifier performs best. To find the subset, the evaluator will use a *search method*, as a linear forward selection method or a best first. The filters use an *attribute evaluator* and a ranker to rank all the features. With this method, each feature, which has lower ranks, is omitted one at a time, to see the accuracy of the classifier. In Weka, one could use *CfsSubSetEval* as an evaluator and *Best First* as a search method. Initially, Weka *Select Attributes* tab was used to evaluate which were the more adequate attribute evaluator and the best search method. For the *attribute selection mode*, there are two possibilities, use full training (FT) set or perform cross-validation (CV).

From the competition dataset with all the 562 (561features + class) features, the steps that were made in order to obtain a 37-feature vector are explained in Table 10 and Table 11:

Table 10: Feature selection from the initial feature vector of the competition dataset (DT0).

Attribute[14]	Attribute Evaluator	Search Method	AttSel Mode	Select Att
561 (DT0)	CfsSubSetEval	BestFirst	FT	49 (DT1)
561 (DT0)	CfsSubSetEval	LinearForwardSel	FT	37 (DT2)
37 (DT2)	CfsSubSetEval	LinearForwardSel	FT	32 (DT3)

[13] In Weka toolkit, features are named as attributes.

[14] Excluding the class feature, as it is the nominal class and is used as the feature for classify.

Table 11: Description of features from the DT 1 and DT 2. Features highlighted in grey were removed in DT 3.

Dataset 1		**Dataset 2**	
Time domain features	*Frequency domain features*	*Time domain features*	*Frequency domain features*
tBodyAccmaxX	fBodyAccstdX	tBodyAccstdX	fBodyAccstdX
tBodyAcccorrelationXY	fBodyAccskewnessX	tBodyAccmaxX	fBodyAccJerkmaxIndsX
tGravityAccmeanX	fBodyAccJerkmaxIndsX	tBodyAccarCoeffZ4	fBodyAccJerkbandsEnergyX18
tGravityAccmeanY	fBodyAccJerkmaxIndsY	tBodyAcccorrelationXY	fBodyAccJerkbandsEnergyZ1724
tGravityAccstdX	fBodyAccJerkmaxIndsZ	tBodyAcccorrelationYZ	fBodyGyromaxIndsX
tGravityAccmaxX	fBodyAccJerkbandsEnergyX18	tGravityAccminX	fBodyGyromaxIndsZ
tGravityAccmaxY	fBodyGyromaxIndsX	tGravityAccminY	fBodyGyromeanFreqX
tGravityAccmaxZ	fBodyGyromaxIndsY	tGravityAccarCoeffX4	fBodyAccMagmean
tGravityAccminX	fBodyGyromaxIndsZ	tGravityAccarCoeffY1	fBodyAccMagmad
tGravityAccminY	fBodyAccMagmean	tGravityAcccorrelationXY	fBodyAccMagiqr
tGravityAccenergyX	fBodyAccMagstd	tGravityAcccorrelationXZ	fBodyBodyAccJerkMagmax
tGravityAccarCoeffX3	fBodyAccMagmad	tBodyAccJerkmadY	fBodyBodyAccJerkMagenergy
tGravityAccarCoeffY1	fBodyAccMagiqr	tBodyAccJerkmaxZ	fBodyBodyGyroMagiqr
tGravityAccarCoeffZ2	fBodyBodyAccJerkMagmax	tBodyAccJerkentropyZ	fBodyBodyGyroMagmaxInds
tGravityAcccorrelationXY	fBodyBodyAccJerkMagenergy	tBodyAccJerkarCoeffX3	angletBodyGyroJerkMeangravityMean
tGravityAcccorrelationYZ	fBodyBodyAccJerkMagiqr	tBodyGyromeanX	
tBodyAccJerkmaxX	fBodyBodyGyroMagmaxInds	tBodyGyroarCoeffY4	
tBodyAccJerkmaxZ	angletBodyGyroMeangravityMean	tBodyGyrocorrelationXY	
tBodyAccJerkiqrZ	angletBodyGyroJerkMeangravityMean	tBodyGyrocorrelationXZ	
tBodyAccJerkentropyZ	angleXgravityMean	tBodyGyrocorrelationYZ	
tBodyAccJerkarCoeffX3	angleYgravityMean	tBodyAccJerkMagmin	
tBodyGyromeanX		tBodyAccJerkMagiqr	
tBodyGyrocorrelationXY			
tBodyGyrocorrelationXZ			
tBodyGyrocorrelationYZ			
tBodyGyroJerkiqrX			
tBodyAccJerkMagmin			
tBodyAccJerkMagiqr			

3.2 Dataset collection

In order to validate the processing developed with the competition dataset, an elderly dataset was collected at Fraunhofer AICOS installations.

The Samsung Galaxy S3 uses a 6-axis IMUs (3D digital accelerometer and 3D digital gyroscope) in iNEMO inertial module (LSM330DLC) with ±2g / ±4g / ±8g / ±16g dynamically selectable full scale (STMicroelectronics, 2012). In the Samsung S3 the maximum range used was ±2g with a resolution of 0.009576807 and power consumption of 0.23mA. In the Samsung

Galaxy S3 mini the inertial module is a MPU-6050ACC with maximum range of ±4g, resolution of 0.15328126 and power consumption of 0.2mA. The sampling rate was not fixed because the phone accelerometer only registers a value when it changes. However it was noticed that Samsung S3 collect more samples per second than the Samsung S3 mini. Pre-processing techniques, as down sampling, were employed to the raw signals acquired with the Smartphones to obtain a 50Hz sampling rate.

The application that records accelerometer data from the Smartphone was already developed in Fraunhofer and it was used in these tests. The objective of this application is to create an annotated dataset. A menu was designed particularly for this tests that includes the four transitions, the four dynamic activities and the circuit explained in Table 13. The supervisors who help elderly performing the tests, needed to select the activity to perform from the menu before its execution and a delay of 5 seconds is provided before starting to record, which enables supervisors to place the phone in the pocket and stabilize it, the recording process is outlined in Figure 11. The Smartphone was always positioned with the screen oriented to the front and its top pointed to the center of the body as represented in Figure 12.

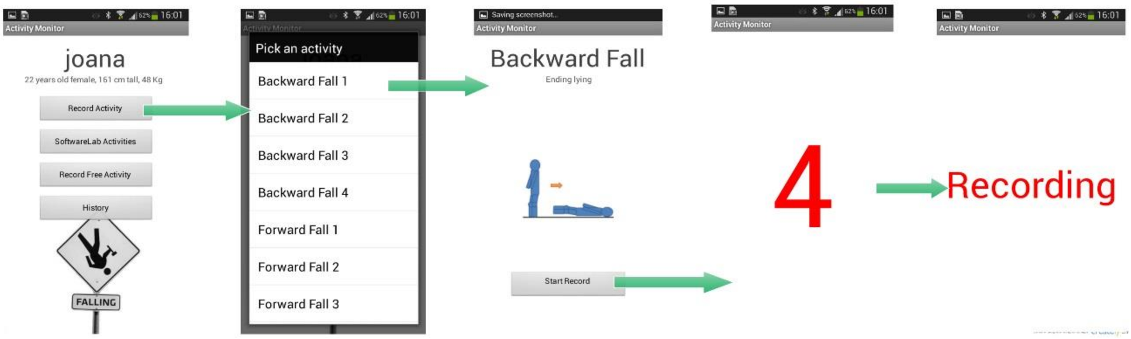


Figure 11: Data recording application already developed from Fraunhofer AICOS.

The tests were carried out with a group of eight elders, four males and four females, with more information detailed in Table 12.

Table 12: Users information. M for male and F for female users.

User	U1	U2	U3	U4	U5	U6	U7	U8	Mean±Std
Age	65	69	70	67	68	67	61	64	66.4 ± 2.7
Weight (Kg)	86	70	67	76	85	70	57	64	71.9 ± 9.4
Height (cm)	170	150	168	170	173	157	156	159	162.9 ± 7.8
Gender	M	F	M	M	M	F	F	F	

The users used a Samsung S3 mini in their left side waist and a Samsung S3 on their right side waist, tied in the user's belt, inside a phone pocket as illustrated in Figure 12.

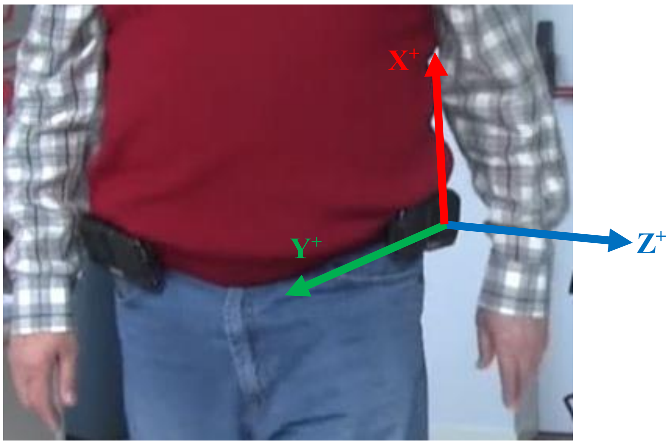


Figure 12: Set-up for the dataset collection with elderly and Smartphone axis orientation, since the Smartphone is placed longitudinally, the axis orientation had rotated 90º from the initial orientation presented in Figure 3.

The activities performed for each participant are described in Table 13 and each one was repeated three times.

Table 13: Activities performing in the dataset recording with elderly.

Activity	Task	Duration (sec)
Transitions	Standing → sit	10
	Sitting → stand	10
	Standing → lay	10
	Lying → stand	10
Dynamic	Normal walk	30
	Fast walk	30
	Up stairs	20
	Down stairs	20
Circuit[15]	stand(5s), sit(5s), stand(5s), walk(10s), lay(10s), upstairs(10s), downstairs(10s)	75

3.3 Threshold-based Approach

The threshold-based approach was studied from the signals obtained with the elderly dataset. This approach allowed to obtain more insights about accelerometer signals during movements and to detect potential features to discriminate activities. The process employed was based in signals supervision and also in literature review.

[15] Consult Annex D for detailed information about the circuit.

3.3.1 Static and Dynamic Activities

The most commonly (Czabke et al., 2011; Feng, Meiling, & Nan, 2011; Karantonis et al., 2006) used method to discriminate between static and dynamic activities is the analysis of the signal magnitude vector (SMV) or the signal magnitude area (SMA). Both metrics conjugate the acceleration in the three axis (*x(i)*, *y(i)* and *z(i)*) into one metric as explain below:

$$\boldsymbol{SMV(i) = \sqrt{x(i)^2 + y(i)^2 + z(i)^2}} \quad (14)$$

$$\boldsymbol{SMA\,(i) = |x(i)| + |y(i)| + |z(i)|} \quad (15)$$

3.3.2 Postural orientation

After determining if a sequence of the signal belongs to static or dynamic group, and concerning the static group, for postural orientation, the differentiation between stand, sit and lay was made using the angle (in degrees) between the accelerometer signal (x, y, z) when the user is in stand position (vector $g = (x_{stand}, y_{stand}, z_{stand})$) and the vector with the accelerometer signal for each instance (vector $v = (x, y, z)$), as explained in equation 16.

$$\boldsymbol{angle(i) = \mathbf{acos}\left(\frac{\vec{g} \bullet \vec{v}(i)}{\|\vec{g}\| \times \|\vec{v}(i)\|}\right) \times \frac{180}{\pi}} \quad (16)$$

This approach requires an initial calibration of the user when he/she is standing, in order to save the vector g, because it is not possible to guarantee that the Smartphone is initially positioned horizontally, and determinate the angle only between the vertical and the vector for each instance. When it is not the case, the initial position of the Smartphone and the vector g will have a component in the three axis and not only on the vertical one.

(Feng et al., 2011) reported that postural transitions could be detected using the maximum and minimum values of the magnitude signal, considering sit-to-stand transition if a local maximum of magnitude appear before a local minimum and considering a stand-to-sit transition if a local minimum of magnitude appear before a local maximum.

3.3.3 Walking and Stairs

Dynamic activities collected signals, as walking and climbing stairs were also analysed in order to develop a method capable of discriminate between then.

The analysis of walking signals was performed in the frequency domain, since this activity has a cyclic pattern which could easily be detected using FFT. Before the FFT application, a high-pass IIR elliptic filter (7th order with 0.25Hz cut-off) (Karantonis et al., 2006) was used to remove the DC component that appears at zero frequency, because the importance of analysing the FFT signals is to detect the first peak frequency in the spectrum, after the zero frequency peak. The peak of interest corresponds to step rate, i.e., the number of steps taken in a time period. The time between each step was also possible to determine when the duration of the test was considered. The analysis was only performed for the vertical axis.

According to (Foerster & Fahrenberg, 2000), walking peak is "within the frequency band of 0.5 to 4 Hz using the z-(longitudinal) axis".

Differentiate between walk and climbing stairs, was not easy using only the graphical inspection and a search of threshold metrics. Nevertheless, it was noticed that walk down stairs had high acceleration variations peaks comparing with walk in level and walk up stairs. This aspect was roughly studied using the standard deviation but the differences between the activities were not significant. In order to solve this problem, an approach based on training with the dynamic activities of the competition dataset and test with the dataset recorded with the elders were employed.

In order to replicate the conditions of the competition dataset to be possible to use the competition dataset to train and the elderly dataset to test, three steps of pre-processing were applied:

1. **Noise filters**

   1.1. Median filter, with 3-length window, applying Matlab function *medfilt.*

   1.2. Butterworth low-pass filter with 20 Hz cut-off frequency and order 3, using Matlab function *butter.*

2. **Separate Body and Gravity components**

   2.1. Butterworth low-pass filter with 0.25 Hz cut-off frequency and order 3. The result of applying this filter is the gravity component (GA) and the body component (BA) result from subtracting the gravity component from the total acceleration signal.

   As mentioned by (Mathie, 2003) the cut-off frequency should be approximately 0.5% the sampling rate, which is in accordance with the values used, with 0.25Hz cut-off frequency for a 50Hz signal.

3. **Sliding windows**

   3.1. From the BA and the GA components, sliding windows of 128 samples with 50% overlap were extracted for each signal.

   For the elderly dataset the class distribution for dynamic activities were: 528 instances for class walk, 165 instances for class up stairs and 159 instances for class down stairs (Table 14). This dataset is not balanced because the number of samples of walking was considerable higher, since the test duration was 30 seconds and the tests for climbing up and down stairs although its duration was 20 seconds people normally take no more than 10 seconds to climb the stairs used in the tests.

Table 14: Datasets class distribution. Highlight values make evident the unbalanced test set.

	Competition dataset, dynamic activities	**Elderly dataset,** dynamic activities
Class	Number of instances	
Walk	1226	528
Walk up stairs	1073	**165**
Walk down stairs	986	**159**

3.4 Summary and Conclusions

The competition dataset was an ideal start up for the study of machine learning concepts applied to AR systems. Although the authors proposed an enormous set of features to extract from the accelerometer signals, through pre-processing techniques it was possible to find a dataset with 38 features which had enough features to consider for a more practical application, such as for a Smartphone AR system.

Using a collected dataset with elderly, it was possible to study critical aspects of accelerometer signal during the execution of different activities. As proposed, it was common practice to reduce a problem with six classes into two sub problems with three classes each, considering static and dynamic activities. Discrimination between activities of each one of the two groups presented several challenges, particularly within dynamic activities (walk and climb stairs).

Comparing with previously developed approaches (Feng et al., 2011; Karantonis et al., 2006), the method described in this project for detecting postural orientation is more precise, since it takes into account that the recording device could not always be positioned in the same orientation, and so requires a user initial calibration step.

For the walking patterns analysis, the approach employed based on FFT was also used before (Foerster & Fahrenberg, 2000; Mathie, 2003) and the methods are suitable, and more analysis could also be done for walking data, as gait monitoring and analysis, in normal or in people with disabilities, or even to track or prevent some diseases. However, the method for discriminating between dynamic activities was not as good as it was expected.

Figure 13 summarizes the methods used in this project to discriminate activities, however not considering the discrimination between walking and climbing stairs due to non-appropriate results. Analysing the SMV of the accelerometer data, if its value was equal to 10m/s^2 the activity was static and the angle feature was used to differentiate different postures. If the activity was dynamic, the angle above 20º and if the SVM was between 5 and 20 m/s^2 the user is considered to be walking if a peak of the FFT spectrum was detect within 1 and 3 Hz, if not a transition could have been occurred. If a minimum of the SMV was found before a maximum a stand-to-sit transition occurred, other else, a sit-to-stand transition occurred.

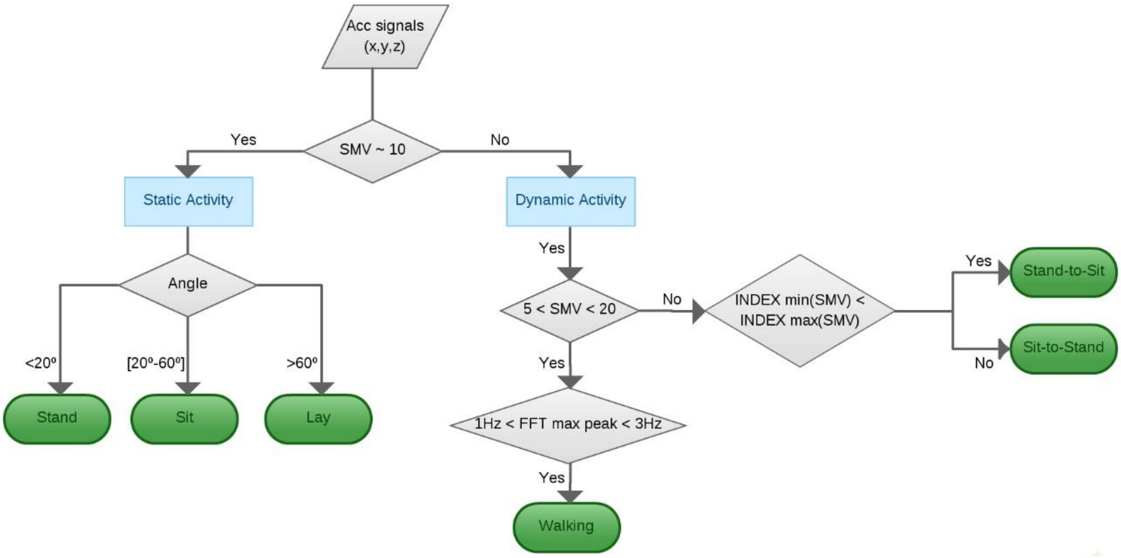


Figure 13: Block diagram of the threshold-based approach.

Chapter 4

Results and Discussion

This chapter summarizes the relevant results obtained with the proposed methods for AR. The results for the machine learning techniques used with the public dataset are explained and the analysis made with the collected dataset was also presented.

4.1 Public Dataset

The results using the competition dataset include the choice of the best dataset, based on selected attributes; the best performance classifier takes into account the number of FP and F-Measure; the performance metrics for the chosen classifier, as accuracy, precision and classifier errors; confusion matrix for a more detailed explanation of the classifier results and the interpretation of the results.

Considering the three datasets obtained with Weka Select Attributes, J48 (pruned tree) was used as the basis classifier, because the interest in this phase is to detect the best reduced dataset. Based on classification performance metrics presented in Table 15, DT2 were chosen as the better reduced dataset, since it presented the highest performance metrics.

Table 15: Classification results for J48 classifier. The values represent the weighted average for the six classes. The highlighted DT2 present the better results.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
DT1	0.84	0.032	0.842	0.84	0.84	0.918
DT2	**0.861**	**0.028**	**0.864**	**0.861**	**0.861**	**0.938**
DT3	0.861	0.028	0.862	0.861	0.86	0.936

Based on Table 16 and considering DT2, IB1 (implementation of a k-NN classifier) and SMO have a statistically significant better performance that J48 classifier, in terms of number of FP and F-Measure. However, J48 is an interpretable classifier and is more suitable for Smartphone implementation. Analysis were made to verify the time needed to train and test the four classifiers presented in Table 16 (results not shown) and comparing to J48, all other classifiers required less time to train but more time to test.

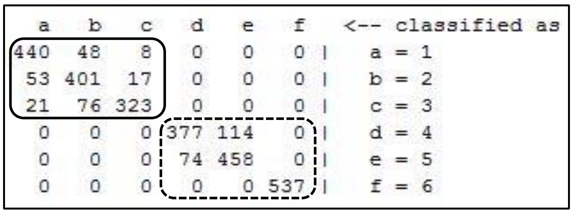
Table 16: Paired Corrected T-Tester with 0.05 confidence (two tailed) for DT2. *v* means that the result is better than the baseline (J48); * means the result is worse than baseline.

	(1) trees.J48	(2) lazy.IB1	(3) bayes.BayesNet	(4) function.SMO
Num. FP	395.20 (v/ /*)	412.20 v (1/0/0)	387.40 * (0/0/1)	407.80 v (1/0/0)
F-Measure	0.94 (v/ /*)	0.98 v (1/0/0)	0.96 (0/1/0)	0.98 v (1/0/0)

Considering DT2 and J48 classifier, Table 17 summarizes the performance metrics obtained for the dataset and the classifier chosen. The obtained accuracy was 86.05% and the classifier error was 13.95%, for a test dataset with 2947 instances and a training dataset with 7352 instances. The corresponding confusion matrix is illustrated in Figure 14. A clear distinction was made between dynamic (full line) and static (dashed) activities, since no static activity were classified as dynamic and vice-versa. In each of the two groups, the diagonal has the high value, i.e., for each activity the majority prediction is for the correspondent class. Lying is the only class with 100% TP.

Table 17: Performance metrics and classifier errors for DT2 and J48 classifier. First row is the accuracy of the classifier and the second row is the classification error. Second column is the number of samples and third column is the percentage.

Metric	**#Samps**	**%**
Correctly Classified Instances	2536	**86,054**
Incorrectly Classified Instances	411	**13,946**
Kappa statistic	0,832	
Mean absolute error	0,048	
Root mean squared error	0,211	
Relative absolute error		17,174
Root relative squared error		56,657
Total Number of Instances	2947	



```
  a   b   c   d   e   f   <-- classified as
440  48   8   0   0   0 |   a = 1
 53 401  17   0   0   0 |   b = 2
 21  76 323   0   0   0 |   c = 3
  0   0   0 377 114   0 |   d = 4
  0   0   0  74 458   0 |   e = 5
  0   0   0   0   0 537 |   f = 6
```

Figure 14: Confusion matrix for DT2. *1- Walking, 2- Walking down stairs, 3- Walking up stairs, 4- Sitting, 5- Standing, 6- Lying.* A clearly distinction was made between dynamic (full line) and static (dashed) activities, since no static activity were classified as dynamic and vice-versa. In each of the two groups, the diagonal has the high value, i.e., for each activity the majority prediction is for the correspondent class. Lying is the only class with 100% TP.

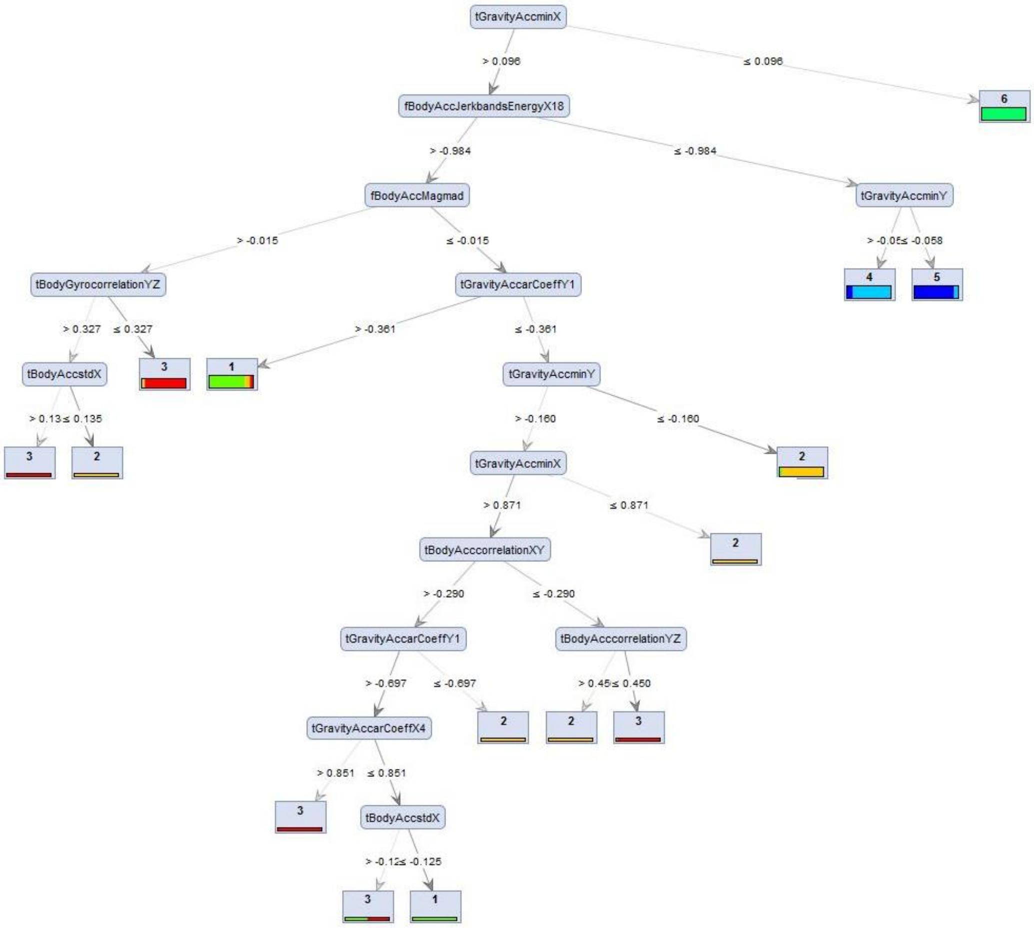


Figure 15: Decision Tree for DT2 using J48 classifier, generated in Rapid Miner[16].

In the results of the Human Activity Recognition Competition for the ESANN 2013 Special Session in "Human Activity and Motion Disorder Recognition", the best performing competitor was University College London with an accuracy of 96.40% (Chiappalone, 2013). The accuracy obtained in this project (86.05%), with J48 classifier (Figure 15), is 10% lower than the winner of the competition, however no information about the process used by the winner was published, so is difficult to conduct a deep comparison of results.

4.2 Elderly Dataset

The results obtained for the threshold-based approach are described below and are divided in four main procedures: (1) differentiate static and dynamic activities based on magnitude values; (2) for static activities, differentiate the postural orientation based on the angle; (3) for dynamic activities, a basic approach for discriminate sit-to-stand and stand-to-sit transitions are presented; (4) analyse walking patterns and detect the number and the duration of a step and an approach for differentiate walking in plane and climbing stairs is proposed.

[16] Rapid Miner was used through the developing of the project to compare results with Weka. The decision tree presented was obtained with Rapid Miner due to better interpretation. However the same conditions were used.

4.2.1 Static and Dynamic Activities

In order to reduce the scope of the classification problem, an initial stage was employed to discriminate static from dynamic activities (Figure 16). As the magnitude signal represent the conjugation of the three components of the acceleration signals, in a static activity the acceleration variations are null and the magnitude has a value around 10m/s^2, which represent the gravity acceleration. When the body moves the orientation vector acquires components in all axis and the magnitude signal deviates from the 10m/s^2, obtaining values upper and down the 10m/s^2 threshold, resultant of the sum of the components having positive or negative values.

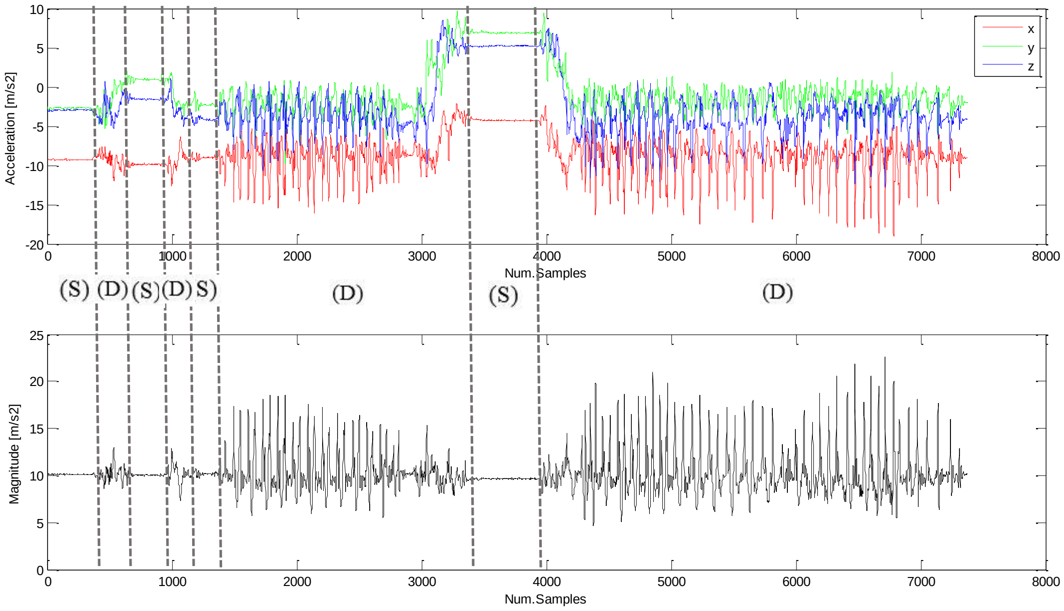


Figure 16: Differentiation between static (S) and dynamic (D) activities was based on magnitude vector (bottom feature). In a static activity the acceleration variations are null and the magnitude has a value around 10m/s^2, which represent the gravity acceleration. When the body moves the orientation vector acquire components in all axis and the magnitude signal deviates from the 10m/s^2, obtaining values upper and down the 10m/s^2 threshold.

4.2.2 Postural orientation

Postural transitions were differentiated using the angle between the initial position of the user and the position of the user during the movement, as represented in Figure 17. Initially when the user were standing, the angle was around 0º; then, the user sited and the angle increases till proximally 20º; the user get up again and the angle returned to 5º (probably the Smartphone moves from the initial position, as the angle in this phase was not 0º); after walking, the user lied down and the angle increased to a value above 85º as expected, since the Smartphone when the user is lying is practically perpendicular (90º) compared to the position of the Smartphone when the user is standing. During walking and climbing stairs, the angle was also above 20º, as the user is upright.

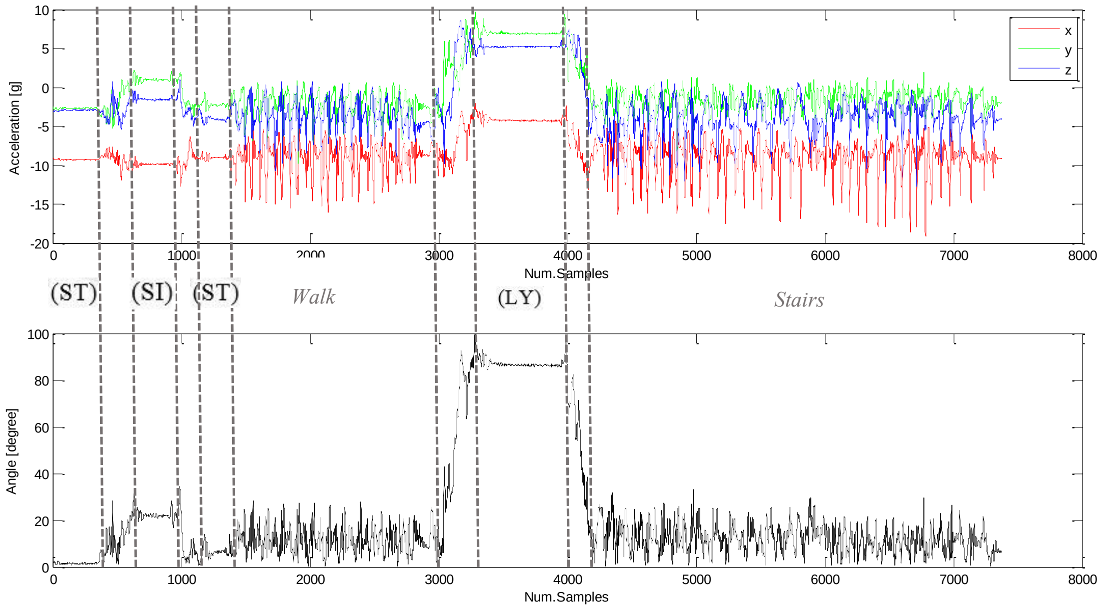


Figure 17: Postural orientation identified with angle (bottom) feature. Below 20º the user is considered in stand position (ST), between 20º and 60º the user is sited (SI) and above 60º the user is laid (LY). Between each posture exists correspondent postural transition.

Postural transitions were roughly identified with angle and magnitude peaks. In Figure 18, the user initially has an angle around 30º, that stands for the sitting position and ends up with an angle around 0º (standing), this indicates that a sit-to-stand transition is implicated between each posture. This transition is highlighted in a dashed line, and detected in the bottom graphic where a maximum peak is represented before a minimum.

The opposite is represented in Figure 19, where the user starts with an angle around 0º and end with an angle above 20º that represented a transition between standing and sitting postures. In the bottom graphic, in the dashed line, a minimum appears before a maximum.

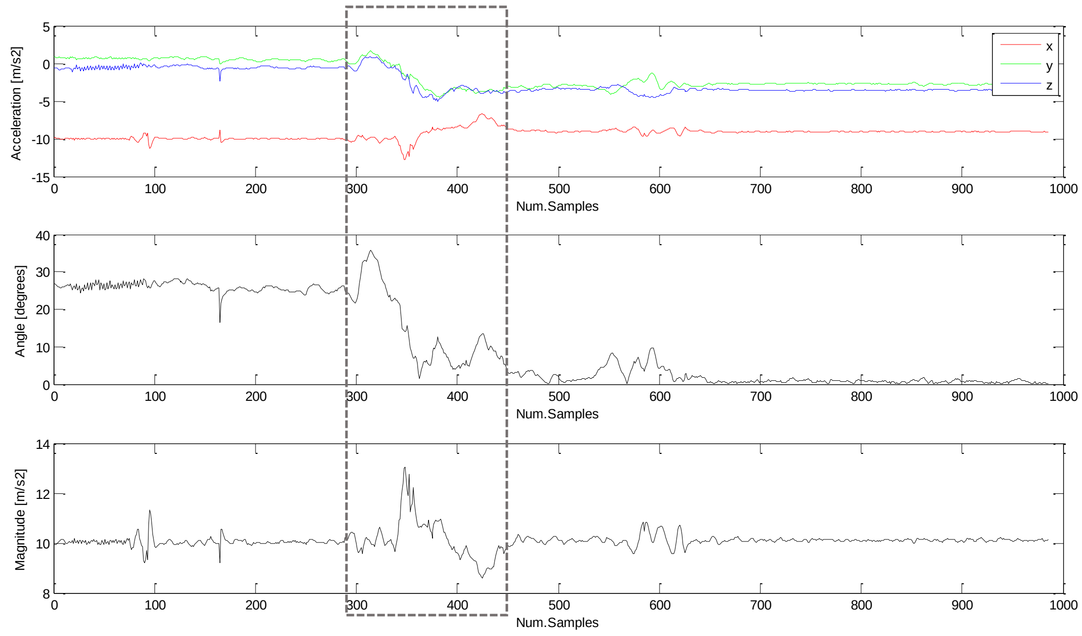


Figure 18: Sit-to-stand transition, identified with angle (middle) and magnitude (bottom) features. Acceleration signals are represented in the top graphic.

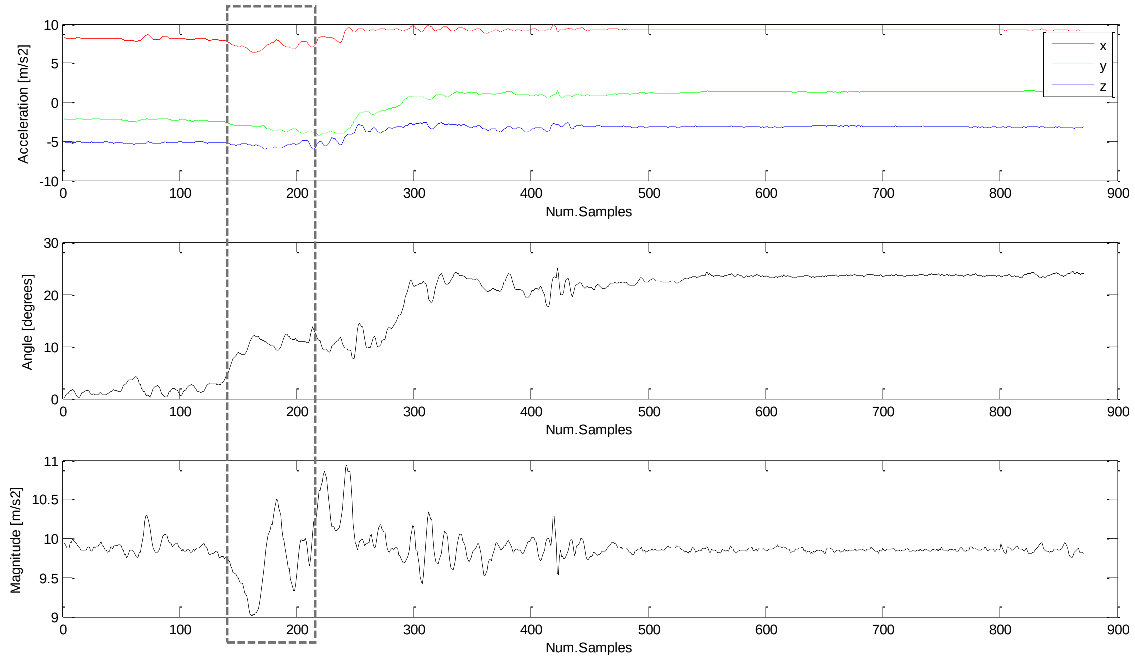


Figure 19: Stand-to-sit transition, identification of the transitions in the accelerometer signal (top) was achieved with angle (middle) and magnitude (bottom) features.

4.2.3 Walking and Stairs

Walking patterns were studied in the frequency domain, using FFT. Figure 20 shows the high pass filter applied before the FFT computation, to remove the DC component around the zero frequency. As represented, the signal initially with an average around 9m/s^2 was transformed in a signal with an average around 0 m/s^2. The peak in the bottom graphic pointed with the arrow result from the initial application of the high pass, but does not change the signal significantly.

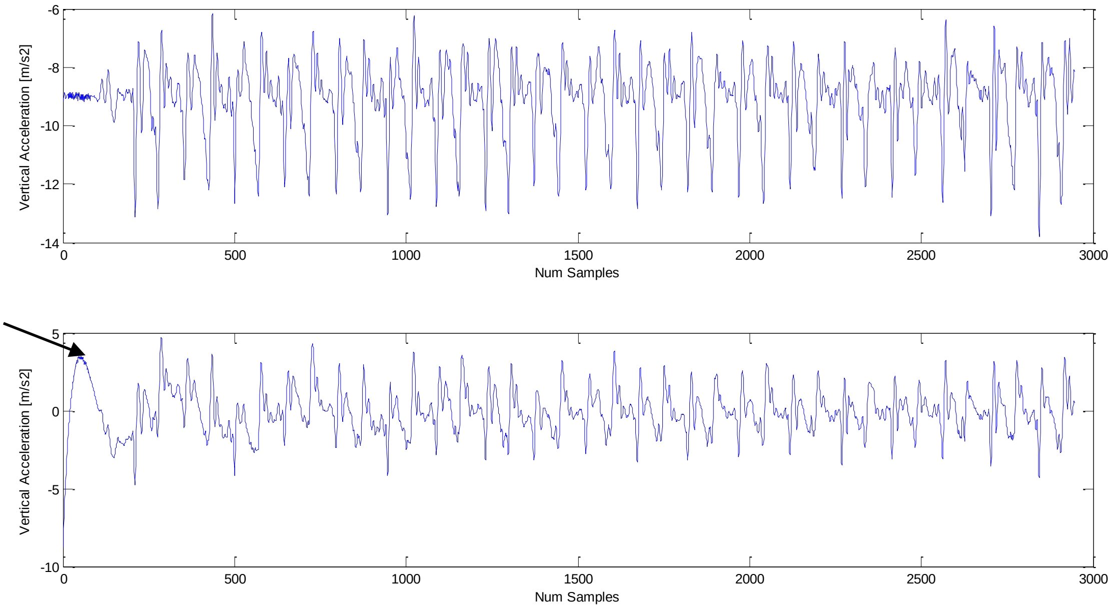


Figure 20: Vertical acceleration during walking after high pass filter (top) and before high pass filter (bottom).

After applying the FFT to the vertical acceleration signal, the step rate is easy to determine, since it corresponds to the peak with the highest amplitude in the frequency domain. A calculus of the number of steps per minute, obtained from a FFT spectrum for a normal walking test, is illustrate in Figure 21, the step rate is 1.343Hz (steps/second), which corresponds to 80 steps/min.

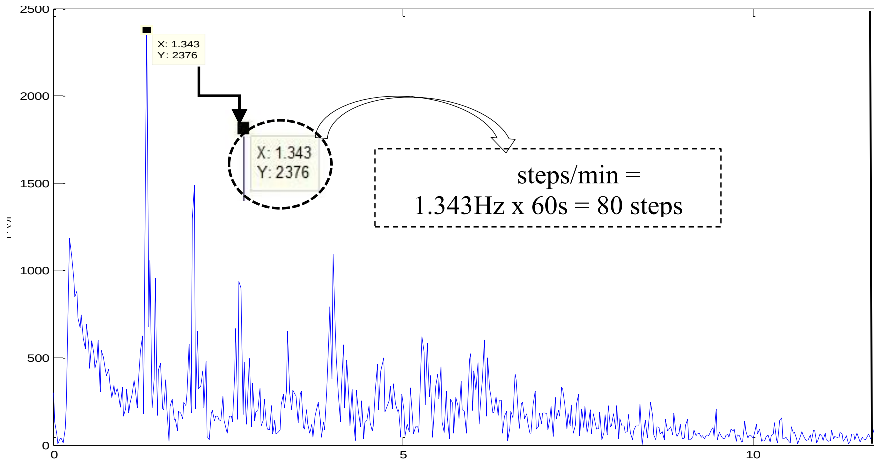


Figure 21: FFT spectrum for a normal walking test, the peak with higher amplitude represent the dominant frequency, i.e., the step rate or the cadence of walking. Since FFT is symmetric in the zero frequency, this spectrum only represent the positive frequencies (x-axis) and the module of the amplitude (y-axis). The result highlighted in the graphic is surrounded in the correspondent sample in **Figure 22**.

The analysis of step rate was made for the normal and fast walking tests. In Figure 22 it is possible to identify that for all of the users the step rate for fast walking is higher than the step rate for the normal walk. In Figure 23 the duration of a double step is represented. Since in the same time interval, a user in fast walking performs more steps per second, the duration of each step should be lower compared to normal walking.

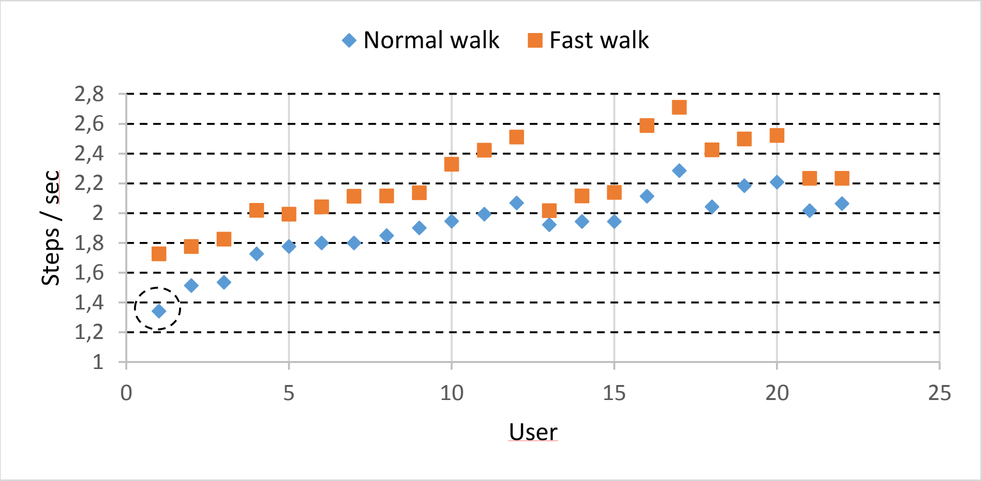


Figure 22: Number of steps per second for normal and fast walk, detected with FFT analysis. Each sample in the graphic corresponds to a user that has two step rates: a lozenge mark for the normal walk and a squared mark for the fast walking.

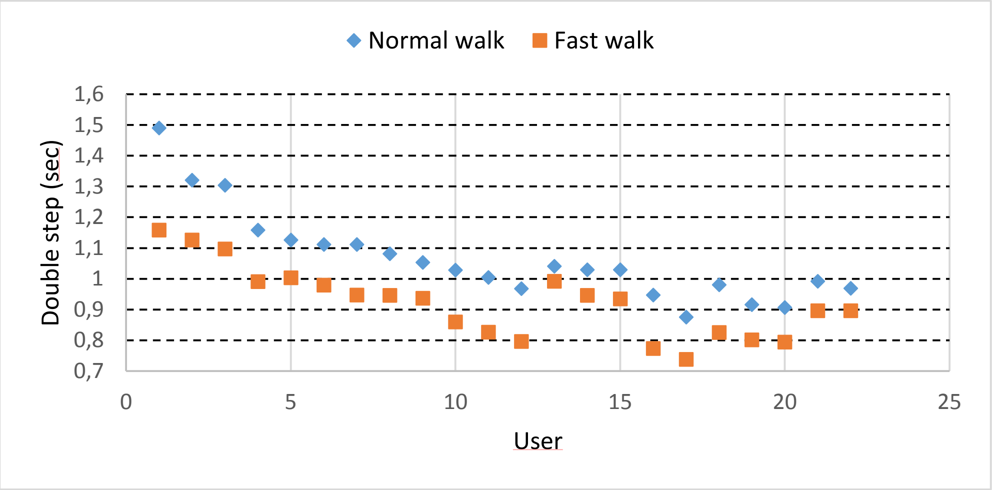


Figure 23: Duration of a double step (in seconds) for normal and fast walk. Each sample in the graphic corresponds to a user that has two step rates: a lozenge mark for the normal walk and a squared mark for the fast walking.

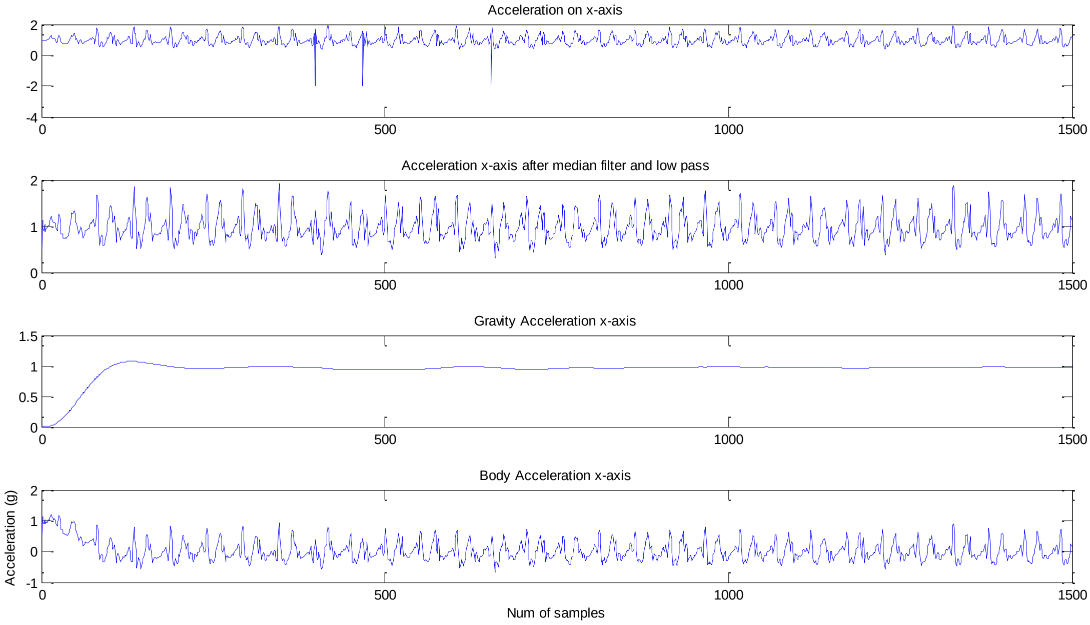


Figure 24: Walking, resulting of applying the pre-processing filters. The second graphic clearly exemplify the removing of noise from acceleration signals (first graphic). Third graphic represents the gravity component of the acceleration signal, the linear acceleration that is constant over the proximally 1g (10m/s^2). The fourth graphic represents the body component of the acceleration signal that contains the acceleration variations during walking.

The approach used to differentiate between walk and climbing stairs, was explained in Section 3.3.3. In order to use the competition dataset to train a classifier that was then used to classify the elderly dataset, the conditions used in the competition were replicated, particular the pre-processing of the raw signals. In Figure 24, the second graphic clearly exemplifies the removing of noise from raw acceleration signals (first graphic). Third graphic represents the gravity component of the acceleration signal, i.e., the linear acceleration, which value is constant over the proximally 1g (10m/s2). The fourth graphic represents the body component of the acceleration signal that contains the acceleration variations during walking. As for the analysis of the competition dataset, first it was necessary to choose the more suitable features to reduce the 531 feature dataset of the competition, taking only into account the dynamic activities. Using the *Select Attributes* in Weka toolkit, it was chosen *InfoGain* as *Attribute Evaluator* and *Ranker* as *Search Method*. The eight best ranked features found were: fBodyAccMagstd, fBodyAccMagmad, fBodyAccMagenergy, tGravityAccMagstd, tBodyAccMagstd, tGravityAccMagmad, tBodyAccMagmad and tBodyAccmaxX. Tests were carried out with datasets with only time domain features and datasets with both time and frequency domain features and the highest performance were obtained for a dataset with only time domain features using trees J48 as classifiers. The results are presented in Table 18 and Table 19.

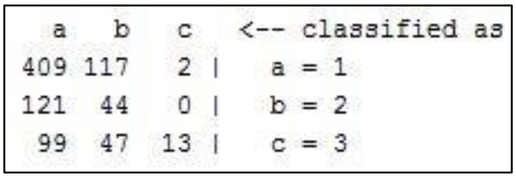
Table 18: Performance metrics for trained J48 with competition dataset (with time-domain features) and tested in elderly dataset.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.775	0.679	0.65	0.775	0.707	0.561	1
	0.267	0.239	0.212	0.267	0.236	0.486	2
	0.082	0.003	0.867	0.082	0.149	0.591	3
Weighted Avg.	**0.547**	**0.468**	**0.606**	**0.547**	**0.512**	**0.552**	

Table 19: Accuracy and classifier errors obtained for the classification scheme explained above. Second column refers to number of samples and third column to percentage

Metric	#Samps	%
Correctly Classified Instances	466	**54,69**
Incorrectly Classified Instances	386	**45,31**
Kappa statistic	0,08	
Mean absolute error	0,31	
Root mean squared error	0,51	
Relative absolute error		71,64
Root relative squared error		111,19
Total Number of Instances	852	

The obtained results are not very good in terms of accuracy, since only almost half of the instances were correctly classified (Table 19). In Figure 25 it is possible to verify that walk was the only class with the higher TP rate, it could be due to the unbalanced test set, which had more walking instances compared with the other two classes. Climbing up and down stairs were mostly classified as walk. Walking up stairs was only confused with walking. A test set with more instances per class, particularly for climbing stairs was necessary in order to obtain better and reliable results. Even if the conditions of the competition dataset in terms of pre-processing were replicated, variation in the process of dataset collection resulting from the used device and the methods employed for recording could have biased the obtained results.



```
  a   b   c   <-- classified as
409 117   2 |   a = 1
121  44   0 |   b = 2
 99  47  13 |   c = 3
```

Figure 25: Confusion matrix obtained for the dynamic dataset. 1-Walk, 2-Walk up stairs and 3-Walk down stairs.

4.3 Summary and Conclusions

The study of the public dataset, using different feature vectors and different classifiers, the most performing approach was a dataset with 38 features used to train a J48 decision tree in Weka. Training that classifier with a dataset of 7352 instances and test in a dataset with 2947 instances, an accuracy of 84% and a classifier error of 14% was obtained.

After collecting an elderly dataset, a threshold-based approach was made in order to get more insights about accelerometer data and to develop a strategy to differentiate activities. First approach was to differentiate between static and dynamic activities using the SMV feature. Within static activities, the angle between the user initial position and the angle of the user in each position was used to recognize postural orientation as standing, sitting and lying. Walking analysis, particularly, step rate and double step duration was evaluated using the FFT spectrum of the vertical acceleration. The peak detected in the spectrum with the higher amplitude corresponds to the step rate.

In order to differentiate walking from climbing stairs, a subset of the public dataset containing only these dynamic activities was used to train a J48 classifier and to test in the collect data, however the results were not good as expected, obtained an accuracy of 54.7%. The differences between the recording methods and the unbalanced test set collected could have biased the results. A solution was to collect more samples from elderly in order to achieve a satisfactory classifier performance.

The bridge between the public and the collected dataset was only made within dynamic activities, since this activities were collected for a duration longer than the static activities, and so obtaining more instances to create a test set. Another drawback is that in the public dataset the static activities consider only the postures and in the collected dataset the static activities consider transitions between them. A pre-processing technique was needed to accurate separate the static postures and even more instances have to be collected.

Chapter 5

Conclusions and Future Work

Human physical activity monitoring has received an increasing interest by elders' caregivers, athletes, physicians, nutritionists, physiotherapists and even people who want to check the daily activity level.

Concerning applications for elderly, and taking into account the actual increasing of aging population and decreasing social and economic conditions for elderly daily care, telecare systems have emerging and have been considered as a solution for some of these problems.

In this project, a study was made regarding the necessary approaches to identify human daily physical activities, such as sitting, lying, stand, walking or climbing stairs.

With reference of the previously studies in this area, Smartphone built-in accelerometers were used to collect motion data from a user performing the activities of interest. The linear acceleration data collected give information about the acceleration due to human body movement and due to the gravity. Extracting metrics from signals, such as magnitude vector, angle, standard deviations of FFT analysis is possible to train a classifier to learn, based on these metrics, how to differentiate activities. Machine learning techniques were used to achieve classifiers training and test. However, a large an accurate dataset is crucial to obtain a reliable classification process.

5.1 Achievements

The objectives of this project regarding the analysis and classification of accelerometers signals were achieved, using two approaches: studying a public and well defined dataset with ADLs obtained from a competition, achieving with a decision tree classifier an accuracy of 86%, 10% below the best performing competitor. The obtained classification error was 14%.

The second approach relied on a collected dataset with elderly voluntaries with ages between 60-70 years. A threshold-based approach was implemented based on collected data, enabling to differentiate static and dynamic activities, and within each group, differentiate postures and walk patterns. The most challenge task was to differentiate between walking in a level and climbing stairs. A tentative to use the public dataset to train a decision tree and test on the collect data was employed, but the results were not revealing, due to considerable reduced test set and the variations of the methods used to collect both datasets.

5.2 Future Work

Further development of this project would be of interest to create a monitor application for human physical activity that could also give a relevant feedback about the user physical history.

The main improvements of this project and future work were identified during its developing and will be explained in detail.

- Explore a wavelet transform approach to discriminate between walking and climbing stairs, as an alternative to the approached followed in this project.

- Study the effect of window size of data streams in the accuracy of the classification, particularly for cyclic activities, that require a larger window size to accurate use FFT approaches.

- Implement the proposed methods on android environment.

- Collect datasets in unsupervised environments, preferentially in a domestic ambient, and use standardized datasets for comparison between classification approaches. The limited uniformed public datasets, make difficult to create an accurate classification algorithm, only if the researcher collects a sufficient number of samples to be able to develop an algorithm. If public standardized databases with human activity could be generated, an investigation considering the differences between activities performed by young and elderly people could also be conducted.

- Enable the user to place the Smartphone in several locations, not only restricted to the waist. In that case, it would be necessary an initial set to calibrate the Smartphone initial position. This improvement could be useful for long term monitoring, but also a disadvantage considering that in some cases the Smartphone moves itself during an activity, due to unfixed support.

- Integrate the Smartphone sensors with other external sensors to improve gait analysis and extend the application to consider physical history, energy expenditure and physical performance evaluations that could give a feedback about the activity level.

- Increase the range of detected activities, to several common sports and more detailed daily living activities, in order to acquire a complete daily history and to be useful for a higher number of users. However, concerning multiple activities has some drawbacks, as the assumption of using the angle to distinct postures, because it is not reliable for activities with external accelerations as sitting in a car or use an elevator.

There are also some open issues considering human activity monitoring, as privacy violation, high FP rates that impossibilities a truthful application of these systems, the high variability within users, that require individual calibration. In terms of market segmentation, there are several recognized applications of these systems, however an important concerning is to create and validate acceptable business models.

References

Anguita, Davide, Ghio, Alessandro, Oneto, Luca, Parra, Xavier, & Reyes-Ortiz, Jorge L. (2012). Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. *International Workshop of Ambient Assisted Living (IWAAL 2012), Vitoria-Gasteiz, Spain*.

Anguita, Davide, Ghio, Alessandro, Oneto, Luca, Parra, Xavier, & Reyes-Ortiz, Jorge L. (2013). A Public Domain Dataset for Human Activity Recognition Using Smartphones. *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*.

Bao, L. (2003). Physical Activity Recognition from Acceleration Data under Semi-Naturalistic Conditions. *M.Eng. Thesis, Massachusetts Institute of Technology*.

Bao, L., & Intille, S. (2004). Activity recognition from user-annotated acceleration data. *Pervasive Computing*, 1-17.

Bird, Steven, Klein, Ewan, & Loper, Edward. (2009). *Natural Language Processing with Python - Chapter 6*: O'Reilly Media.

Campilho, Aurélio. (2009a). Computer Aided Diagnosis: Methodologies Feature Extraction and Classification.

Campilho, Aurélio. (2009b). Nonparametric learning. *Computer-aided diagnosis, Mestrado Integrado em Biongenharia, FEUP*.

Chiappalone, Marco. (2013). http://www.smartlab.ws/component/content/article?layout=edit&id=61. Retrieved 25-06-2013

CreatioSoft. (2012-2013). http://creatiosoft.com/forum/showthread.php?tid=13. Retrieved 24-06-2013

Czabke, A., Marsch, S., & Lueth, T. C. (2011, 23-26 May 2011). *Accelerometer based real-time activity analysis on a microcontroller.* Paper presented at the Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on.

Feng, Wang, Meiling, Wang, & Nan, Feng. (2011, 24-28 Sept. 2011). *Research on Classification of Human Daily Activities Based on a Single Tri-Axial Accelerometer.* Paper presented at the Complexity and Data Mining (IWCDM), 2011 First International Workshop on.

Figo, D., Diniz, P. C., Ferreira, D. R., & Cardoso, J. M. P. (2010). Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing, 14*(7), 645-662.

Foerster, Friedrich, & Fahrenberg, Jochen. (2000). Motion pattern and posture: Correctly assessed by calibrated accelerometers. *Behavior Research Methods, Instruments, & Computers, 32*(3), 450-457. doi: 10.3758/bf03200815

Fraunhofer. (2013a). http://smartcompanion.projects.fraunhofer.pt/. Retrieved 22-06-2013

Fraunhofer. (2013b). http://www.fraunhofer.pt/en/fraunhofer_aicos/about_us.html. Retrieved 16-06-2013

Gomes, J., Krishnaswamy, S., Gaber, Mohamed, Sousa, P. , & Menasalvas, E. (2012). *MARS: a personalised Mobile Activity Recognition System.* Paper presented at the Proceedings of the IEEE International Conference on Mobile Data Management, Bengaluru, India.

Huynh, Duy Tam Gilles. (2008). Human Activity Recognition with Wearable Sensors. *Dissertation*.

Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H., & Celler, B. G. (2006). Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *Trans. Info. Tech. Biomed., 10*(1), 156-167. doi: 10.1109/titb.2005.856864

Karjalainen, Sampo. (2013). http://www.moves-app.com/. Retrieved 16-06-2013

Kuuti, K. (1995). Activity Theory as a potential framework for human-computer interaction research framework. *Context and Consciousness: Activity Theory and Human Computer Interaction (Nardi, Bonnie A., ed.), 1/1*(2), 17-44.

Lara, O., & Labrador, M. (2012). A Survey on Human Activity Recognition using Wearable Sensors. *Communications Surveys & Tutorials, IEEE, PP*(99), 1-18. doi: 10.1109/surv.2012.110112.00192

Lee, Myong-Woo, Khan, Adil Mehmood, & Kim, Tae-Seong. (2011). A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation. *Personal Ubiquitous Comput., 15*(8), 887-898. doi: 10.1007/s00779-011-0403-3

Lopes, A., Mendes-Moreira, J., & Gama, J. (2012). Semi-supervised learning: predicting activities in Android environment. *Workshop on Ubiquitous Data*, 1-5.

Mathie, Merryn Joy. (2003). *Monitoring and interpreting human movement patterns using a triaxial accelerometer.* (PhD), Faculty of Engineering, UNSW, Australia.

Maurer, U., Smailagic, A., Siewiorek, D. P., & Deisher, M. (2006, 3-5 April 2006). *Activity recognition and monitoring using multiple sensors on different body positions.* Paper presented at the Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on.

Mendonça, Ana Maria. (2009). Evaluation and Interpretation of Computer Aided Diagnosis Systems. *Computer-aided diagnosis, Mestrado Integrado em Biongenharia, FEUP*.

Nadales, María. (2010). *Recognition of human motion related activities from sensors.* (Msc), Universidad de Málaga, Málaga.

Qiang, Li, Stankovic, J. A., Hanson, M. A., Barth, A. T., Lach, J., & Gang, Zhou. (2009, 3-5 June 2009). *Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information.* Paper presented at the Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on.

Research, Shimmer.). http://www.shimmer-research.com/applications-2/pamsys-physical-activity-monitoring. Retrieved 18-06-2013

Riboni, D., & Bettini, C. (2010). COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*(1-19).

Ristic, Lj. (2012). Sensor fusion and MEMS for 10-DoF solutions. *Petrov Group, EE Times, Design, Medical*.

Rubenstein, Laurence Z., & Josephson, Karen R. (2006). Falls and Their Prevention in Elderly People: What Does the Evidence Show? *Medical Clinics of North America, 90*(5), 807-824. doi: http://dx.doi.org/10.1016/j.mcna.2006.05.013

Saponas, T. Scott, Lester, Jonathan, Froehlich, Jon E., Fogarty, James, & Landay, James A. (2008). iLearn on the iPhone: Real-Time Human Activity Classification on Commodity Mobile Phones.

STMicroelectronics. (2012). LSM330DLC Datasheet.

Ugulino, Wallace, Velloso, Eduardo, Milidiú, Ruy Luiz, & Fuks, Hugo. (2012). *Human Activity Recognition using On-body Sensing.* Paper presented at the Proceedings of III Symposium of the Brazilian Institute for Web Science Research (WebScience).

Wilde, A. (2010). An overview of human activity detection technologies for pervasive systems. *Seminar paper. Pervasive and Artificial Intelligence Group of University of Southampton*.

Wilde, A. (2011). Activity recognition for motion-aware pervasive systems. *Master Thesis of University of Southampton*.

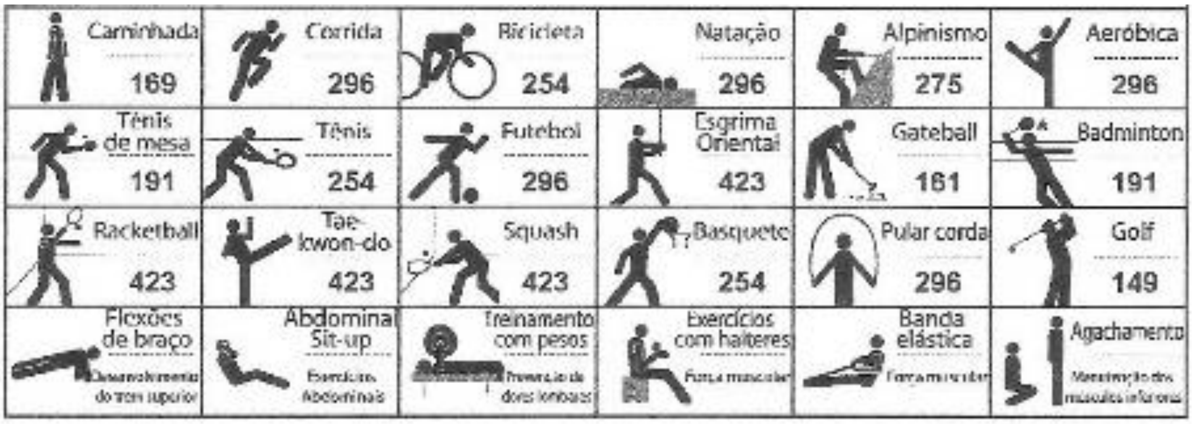
Yu, Robert. (2013). Analog, MEMS and Sensors Enable our Mobile Devices Into a SMART World.

Zhou, Zhi-Hua. Ensemble Learning. *National Key Laboratory for Novel Software Technology, Nanjing University, China*.

Annex A

Nutritional counselling

The human activity monitoring and the nutritional counselling have a straight relationship, since every time people need to lose weight the nutritionist recommends a diet and also a physical plan. In Figure 26 is illustrated an example of a physical plan used in a nutritional clinic. Activity recognition systems could be useful in this situations to check if the user is performing the recommended physical plan. However, attention should be taken in the sports detection accuracy, which could be difficult in same cases to discriminate.



Caminhada 169	Corrida 296	Bicicleta 254	Natação 296	Alpinismo 275	Aeróbica 296
Ténis de mesa 191	Ténis 254	Futebol 296	Esgrima Oriental 423	Gateball 161	Badminton 191
Racketball 423	Tae-kwon-do 423	Squash 423	Basquete 254	Pular corda 296	Golf 149
Flexões de braço Desenvolvimento do tronco superior	Abdominal Sit-up Exercícios Abdominais	Treinamento com pesos Prevenção de dores lombares	Exercícios com halteres Força muscular	Banda elástica Força muscular	Agachamento Manutenção dos músculos inferiores

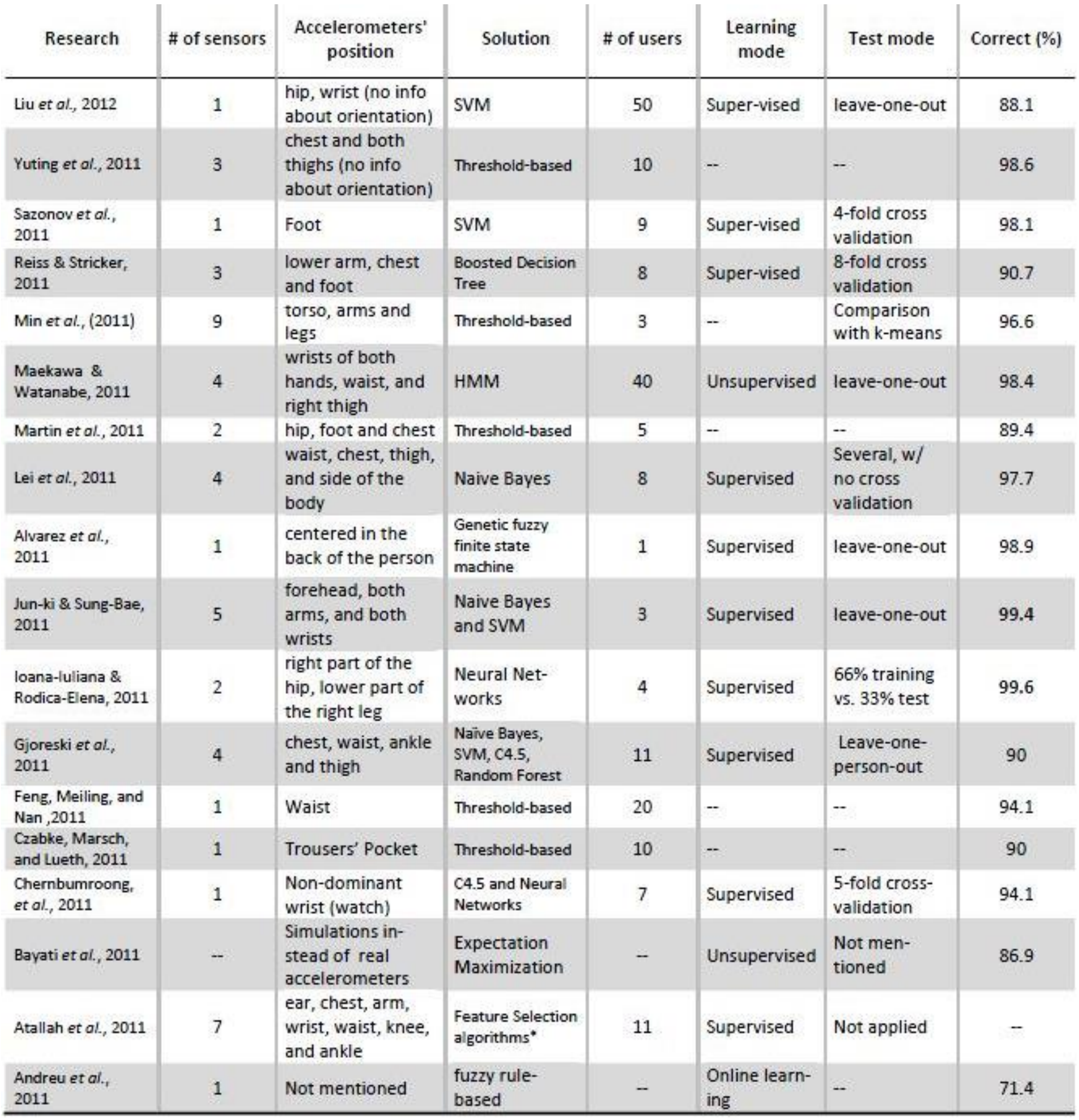
Figure 26: Exercise plan used in a nutritional clinic. The energetic expenditure for each activity is calculated for a male adult with 84.5Kg. The energetic expenditure is calculated in Kcal for an activity duration of 30minutes. Source: *Clínica Dr. Fernando Póvoas, Porto*.

Annex B

HAR state of the art

Based on IEEE database from 2012 and 2011(Ugulino et al., 2012), the Human Activity Recognition (HAR) studies (Table 20) based on accelerometers are the following:

Table 20: State of the art HAR studies (2012 and 2011).



Research	# of sensors	Accelerometers' position	Solution	# of users	Learning mode	Test mode	Correct (%)
Liu *et al.*, 2012	1	hip, wrist (no info about orientation)	SVM	50	Super-vised	leave-one-out	88.1
Yuting *et al.*, 2011	3	chest and both thighs (no info about orientation)	Threshold-based	10	--	--	98.6
Sazonov *et al.*, 2011	1	Foot	SVM	9	Super-vised	4-fold cross validation	98.1
Reiss & Stricker, 2011	3	lower arm, chest and foot	Boosted Decision Tree	8	Super-vised	8-fold cross validation	90.7
Min *et al.*, (2011)	9	torso, arms and legs	Threshold-based	3	--	Comparison with k-means	96.6
Maekawa & Watanabe, 2011	4	wrists of both hands, waist, and right thigh	HMM	40	Unsupervised	leave-one-out	98.4
Martin *et al.*, 2011	2	hip, foot and chest	Threshold-based	5	--	--	89.4
Lei *et al.*, 2011	4	waist, chest, thigh, and side of the body	Naive Bayes	8	Supervised	Several, w/ no cross validation	97.7
Alvarez *et al.*, 2011	1	centered in the back of the person	Genetic fuzzy finite state machine	1	Supervised	leave-one-out	98.9
Jun-ki & Sung-Bae, 2011	5	forehead, both arms, and both wrists	Naive Bayes and SVM	3	Supervised	leave-one-out	99.4
Ioana-Iuliana & Rodica-Elena, 2011	2	right part of the hip, lower part of the right leg	Neural Networks	4	Supervised	66% training vs. 33% test	99.6
Gjoreski *et al.*, 2011	4	chest, waist, ankle and thigh	Naïve Bayes, SVM, C4.5, Random Forest	11	Supervised	Leave-one-person-out	90
Feng, Meiling, and Nan ,2011	1	Waist	Threshold-based	20	--	--	94.1
Czabke, Marsch, and Lueth, 2011	1	Trousers' Pocket	Threshold-based	10	--	--	90
Chernbumroong, *et al.*, 2011	1	Non-dominant wrist (watch)	C4.5 and Neural Networks	7	Supervised	5-fold cross-validation	94.1
Bayati *et al.*, 2011	--	Simulations instead of real accelerometers	Expectation Maximization	--	Unsupervised	Not mentioned	86.9
Atallah *et al.*, 2011	7	ear, chest, arm, wrist, waist, knee, and ankle	Feature Selection algorithms*	11	Supervised	Not applied	--
Andreu *et al.*, 2011	1	Not mentioned	fuzzy rule-based	--	Online learning	--	71.4

Annex C

Competition Dataset Features

This annex details the collection of the 561 features that (Anguita et al., 2013) had extracted and result from 17 features extracted over 17 signals (some of the signals have 3D components).

1 tBodyAcc-mean()-X
2 tBodyAcc-mean()-Y
3 tBodyAcc-mean()-Z
4 tBodyAcc-std()-X
5 tBodyAcc-std()-Y
6 tBodyAcc-std()-Z
7 tBodyAcc-mad()-X
8 tBodyAcc-mad()-Y
9 tBodyAcc-mad()-Z
10 tBodyAcc-max()-X
11 tBodyAcc-max()-Y
12 tBodyAcc-max()-Z
13 tBodyAcc-min()-X
14 tBodyAcc-min()-Y
15 tBodyAcc-min()-Z
16 tBodyAcc-sma()
17 tBodyAcc-energy()-X
18 tBodyAcc-energy()-Y
19 tBodyAcc-energy()-Z
20 tBodyAcc-iqr()-X
21 tBodyAcc-iqr()-Y
22 tBodyAcc-iqr()-Z
23 tBodyAcc-entropy()-X
24 tBodyAcc-entropy()-Y
25 tBodyAcc-entropy()-Z
26 tBodyAcc-arCoeff()-X,1
27 tBodyAcc-arCoeff()-X,2
28 tBodyAcc-arCoeff()-X,3
29 tBodyAcc-arCoeff()-X,4
30 tBodyAcc-arCoeff()-Y,1
31 tBodyAcc-arCoeff()-Y,2
32 tBodyAcc-arCoeff()-Y,3
33 tBodyAcc-arCoeff()-Y,4
34 tBodyAcc-arCoeff()-Z,1
201 tBodyAccMag-mean()
202 tBodyAccMag-std()
203 tBodyAccMag-mad()
204 tBodyAccMag-max()
205 tBodyAccMag-min()
206 tBodyAccMag-sma()
207 tBodyAccMag-energy()
208 tBodyAccMag-iqr()
209 tBodyAccMag-entropy()
210 tBodyAccMag-arCoeff()1
211 tBodyAccMag-arCoeff()2
212 tBodyAccMag-arCoeff()3
213 tBodyAccMag-arCoeff()4
214 tGravityAccMag-mean()
215 tGravityAccMag-std()
216 tGravityAccMag-mad()
217 tGravityAccMag-max()
218 tGravityAccMag-min()
219 tGravityAccMag-sma()
220 tGravityAccMag-energy()
221 tGravityAccMag-iqr()
222 tGravityAccMag-entropy()
223 tGravityAccMag-arCoeff()1
224 tGravityAccMag-arCoeff()2
225 tGravityAccMag-arCoeff()3
226 tGravityAccMag-arCoeff()4
227 tBodyAccJerkMag-mean()
228 tBodyAccJerkMag-std()
229 tBodyAccJerkMag-mad()
230 tBodyAccJerkMag-max()
231 tBodyAccJerkMag-min()
232 tBodyAccJerkMag-sma()
233 tBodyAccJerkMag-energy()
234 tBodyAccJerkMag-iqr()
401 fBodyAccJerk-bandsEnergy()-41,48y
402 fBodyAccJerk-bandsEnergy()-49,56y
403 fBodyAccJerk-bandsEnergy()-57,64y
404 fBodyAccJerk-bandsEnergy()-1,16y
405 fBodyAccJerk-bandsEnergy()-17,32y
406 fBodyAccJerk-bandsEnergy()-33,48y
407 fBodyAccJerk-bandsEnergy()-49,64y
408 fBodyAccJerk-bandsEnergy()-1,24y
409 fBodyAccJerk-bandsEnergy()-25,48y
410 fBodyAccJerk-bandsEnergy()-1,8z
411 fBodyAccJerk-bandsEnergy()-9,16z
412 fBodyAccJerk-bandsEnergy()-17,24z
413 fBodyAccJerk-bandsEnergy()-25,32z
414 fBodyAccJerk-bandsEnergy()-33,40z
415 fBodyAccJerk-bandsEnergy()-41,48z
416 fBodyAccJerk-bandsEnergy()-49,56z
417 fBodyAccJerk-bandsEnergy()-57,64z
418 fBodyAccJerk-bandsEnergy()-1,16z
419 fBodyAccJerk-bandsEnergy()-17,32z
420 fBodyAccJerk-bandsEnergy()-33,48z
421 fBodyAccJerk-bandsEnergy()-49,64z
422 fBodyAccJerk-bandsEnergy()-1,24z
423 fBodyAccJerk-bandsEnergy()-25,48z
424 fBodyGyro-mean()-X
425 fBodyGyro-mean()-Y
426 fBodyGyro-mean()-Z
427 fBodyGyro-std()-X
428 fBodyGyro-std()-Y
429 fBodyGyro-std()-Z
430 fBodyGyro-mad()-X
431 fBodyGyro-mad()-Y
432 fBodyGyro-mad()-Z
433 fBodyGyro-max()-X
434 fBodyGyro-max()-Y

35 tBodyAcc-arCoeff()-Z,2
36 tBodyAcc-arCoeff()-Z,3
37 tBodyAcc-arCoeff()-Z,4
38 tBodyAcc-correlation()-X,Y
39 tBodyAcc-correlation()-X,Z
40 tBodyAcc-correlation()-Y,Z
41 tGravityAcc-mean()-X
42 tGravityAcc-mean()-Y
43 tGravityAcc-mean()-Z
44 tGravityAcc-std()-X
45 tGravityAcc-std()-Y
46 tGravityAcc-std()-Z
47 tGravityAcc-mad()-X
48 tGravityAcc-mad()-Y
49 tGravityAcc-mad()-Z
50 tGravityAcc-max()-X
51 tGravityAcc-max()-Y
52 tGravityAcc-max()-Z
53 tGravityAcc-min()-X
54 tGravityAcc-min()-Y
55 tGravityAcc-min()-Z
56 tGravityAcc-sma()
57 tGravityAcc-energy()-X
58 tGravityAcc-energy()-Y
59 tGravityAcc-energy()-Z
60 tGravityAcc-iqr()-X
61 tGravityAcc-iqr()-Y
62 tGravityAcc-iqr()-Z
63 tGravityAcc-entropy()-X
64 tGravityAcc-entropy()-Y
65 tGravityAcc-entropy()-Z
66 tGravityAcc-arCoeff()-X,1
67 tGravityAcc-arCoeff()-X,2
68 tGravityAcc-arCoeff()-X,3
69 tGravityAcc-arCoeff()-X,4
70 tGravityAcc-arCoeff()-Y,1
71 tGravityAcc-arCoeff()-Y,2
72 tGravityAcc-arCoeff()-Y,3
73 tGravityAcc-arCoeff()-Y,4
74 tGravityAcc-arCoeff()-Z,1
75 tGravityAcc-arCoeff()-Z,2
76 tGravityAcc-arCoeff()-Z,3
77 tGravityAcc-arCoeff()-Z,4
78 tGravityAcc-correlation()-X,Y
79 tGravityAcc-correlation()-X,Z
80 tGravityAcc-correlation()-Y,Z

235 tBodyAccJerkMag-entropy()
236 tBodyAccJerkMag-arCoeff()1
237 tBodyAccJerkMag-arCoeff()2
238 tBodyAccJerkMag-arCoeff()3
239 tBodyAccJerkMag-arCoeff()4
240 tBodyGyroMag-mean()
241 tBodyGyroMag-std()
242 tBodyGyroMag-mad()
243 tBodyGyroMag-max()
244 tBodyGyroMag-min()
245 tBodyGyroMag-sma()
246 tBodyGyroMag-energy()
247 tBodyGyroMag-iqr()
248 tBodyGyroMag-entropy()
249 tBodyGyroMag-arCoeff()1
250 tBodyGyroMag-arCoeff()2
251 tBodyGyroMag-arCoeff()3
252 tBodyGyroMag-arCoeff()4
253 tBodyGyroJerkMag-mean()
254 tBodyGyroJerkMag-std()
255 tBodyGyroJerkMag-mad()
256 tBodyGyroJerkMag-max()
257 tBodyGyroJerkMag-min()
258 tBodyGyroJerkMag-sma()
259 tBodyGyroJerkMag-energy()
260 tBodyGyroJerkMag-iqr()
261 tBodyGyroJerkMag-entropy()
262 tBodyGyroJerkMag-arCoeff()1
263 tBodyGyroJerkMag-arCoeff()2
264 tBodyGyroJerkMag-arCoeff()3
265 tBodyGyroJerkMag-arCoeff()4
266 fBodyAcc-mean()-X
267 fBodyAcc-mean()-Y
268 fBodyAcc-mean()-Z
269 fBodyAcc-std()-X
270 fBodyAcc-std()-Y
271 fBodyAcc-std()-Z
272 fBodyAcc-mad()-X
273 fBodyAcc-mad()-Y
274 fBodyAcc-mad()-Z
275 fBodyAcc-max()-X
276 fBodyAcc-max()-Y
277 fBodyAcc-max()-Z
278 fBodyAcc-min()-X
279 fBodyAcc-min()-Y
280 fBodyAcc-min()-Z

435 fBodyGyro-max()-Z
436 fBodyGyro-min()-X
437 fBodyGyro-min()-Y
438 fBodyGyro-min()-Z
439 fBodyGyro-sma()
440 fBodyGyro-energy()-X
441 fBodyGyro-energy()-Y
442 fBodyGyro-energy()-Z
443 fBodyGyro-iqr()-X
444 fBodyGyro-iqr()-Y
445 fBodyGyro-iqr()-Z
446 fBodyGyro-entropy()-X
447 fBodyGyro-entropy()-Y
448 fBodyGyro-entropy()-Z
449 fBodyGyro-maxInds-X
450 fBodyGyro-maxInds-Y
451 fBodyGyro-maxInds-Z
452 fBodyGyro-meanFreq()-X
453 fBodyGyro-meanFreq()-Y
454 fBodyGyro-meanFreq()-Z
455 fBodyGyro-skewness()-X
456 fBodyGyro-kurtosis()-X
457 fBodyGyro-skewness()-Y
458 fBodyGyro-kurtosis()-Y
459 fBodyGyro-skewness()-Z
460 fBodyGyro-kurtosis()-Z
461 fBodyGyro-bandsEnergy()-1,8x
462 fBodyGyro-bandsEnergy()-9,16x
463 fBodyGyro-bandsEnergy()-17,24x
464 fBodyGyro-bandsEnergy()-25,32x
465 fBodyGyro-bandsEnergy()-33,40x
466 fBodyGyro-bandsEnergy()-41,48x
467 fBodyGyro-bandsEnergy()-49,56x
468 fBodyGyro-bandsEnergy()-57,64x
469 fBodyGyro-bandsEnergy()-1,16x
470 fBodyGyro-bandsEnergy()-17,32x
471 fBodyGyro-bandsEnergy()-33,48x
472 fBodyGyro-bandsEnergy()-49,64x
473 fBodyGyro-bandsEnergy()-1,24x
474 fBodyGyro-bandsEnergy()-25,48x
475 fBodyGyro-bandsEnergy()-1,8y
476 fBodyGyro-bandsEnergy()-9,16y
477 fBodyGyro-bandsEnergy()-17,24y
478 fBodyGyro-bandsEnergy()-25,32y
479 fBodyGyro-bandsEnergy()-33,40y
480 fBodyGyro-bandsEnergy()-41,48y

81 tBodyAccJerk-mean()-X
82 tBodyAccJerk-mean()-Y
83 tBodyAccJerk-mean()-Z
84 tBodyAccJerk-std()-X
85 tBodyAccJerk-std()-Y
86 tBodyAccJerk-std()-Z
87 tBodyAccJerk-mad()-X
88 tBodyAccJerk-mad()-Y
89 tBodyAccJerk-mad()-Z
90 tBodyAccJerk-max()-X
91 tBodyAccJerk-max()-Y
92 tBodyAccJerk-max()-Z
93 tBodyAccJerk-min()-X
94 tBodyAccJerk-min()-Y
95 tBodyAccJerk-min()-Z
96 tBodyAccJerk-sma()
97 tBodyAccJerk-energy()-X
98 tBodyAccJerk-energy()-Y
99 tBodyAccJerk-energy()-Z
100 tBodyAccJerk-iqr()-X
101 tBodyAccJerk-iqr()-Y
102 tBodyAccJerk-iqr()-Z
103 tBodyAccJerk-entropy()-X
104 tBodyAccJerk-entropy()-Y
105 tBodyAccJerk-entropy()-Z
106 tBodyAccJerk-arCoeff()-X,1
107 tBodyAccJerk-arCoeff()-X,2
108 tBodyAccJerk-arCoeff()-X,3
109 tBodyAccJerk-arCoeff()-X,4
110 tBodyAccJerk-arCoeff()-Y,1
111 tBodyAccJerk-arCoeff()-Y,2
112 tBodyAccJerk-arCoeff()-Y,3
113 tBodyAccJerk-arCoeff()-Y,4
114 tBodyAccJerk-arCoeff()-Z,1
115 tBodyAccJerk-arCoeff()-Z,2
116 tBodyAccJerk-arCoeff()-Z,3
117 tBodyAccJerk-arCoeff()-Z,4
118 tBodyAccJerk-correlation()-X,Y
119 tBodyAccJerk-correlation()-X,Z
120 tBodyAccJerk-correlation()-Y,Z
121 tBodyGyro-mean()-X
122 tBodyGyro-mean()-Y
123 tBodyGyro-mean()-Z
124 tBodyGyro-std()-X
125 tBodyGyro-std()-Y

281 fBodyAcc-sma()
282 fBodyAcc-energy()-X
283 fBodyAcc-energy()-Y
284 fBodyAcc-energy()-Z
285 fBodyAcc-iqr()-X
286 fBodyAcc-iqr()-Y
287 fBodyAcc-iqr()-Z
288 fBodyAcc-entropy()-X
289 fBodyAcc-entropy()-Y
290 fBodyAcc-entropy()-Z
291 fBodyAcc-maxInds-X
292 fBodyAcc-maxInds-Y
293 fBodyAcc-maxInds-Z
294 fBodyAcc-meanFreq()-X
295 fBodyAcc-meanFreq()-Y
296 fBodyAcc-meanFreq()-Z
297 fBodyAcc-skewness()-X
298 fBodyAcc-kurtosis()-X
299 fBodyAcc-skewness()-Y
300 fBodyAcc-kurtosis()-Y
301 fBodyAcc-skewness()-Z
302 fBodyAcc-kurtosis()-Z
303 fBodyAcc-bandsEnergy()-1,8x
304 fBodyAcc-bandsEnergy()-9,16x
305 fBodyAcc-bandsEnergy()-17,24x
306 fBodyAcc-bandsEnergy()-25,32x
307 fBodyAcc-bandsEnergy()-33,40x
308 fBodyAcc-bandsEnergy()-41,48x
309 fBodyAcc-bandsEnergy()-49,56x
310 fBodyAcc-bandsEnergy()-57,64x
311 fBodyAcc-bandsEnergy()-1,16x
312 fBodyAcc-bandsEnergy()-17,32x
313 fBodyAcc-bandsEnergy()-33,48x
314 fBodyAcc-bandsEnergy()-49,64x
315 fBodyAcc-bandsEnergy()-1,24x
316 fBodyAcc-bandsEnergy()-25,48x
317 fBodyAcc-bandsEnergy()-1,8y
318 fBodyAcc-bandsEnergy()-9,16y
319 fBodyAcc-bandsEnergy()-17,24y
320 fBodyAcc-bandsEnergy()-25,32y
321 fBodyAcc-bandsEnergy()-33,40y
322 fBodyAcc-bandsEnergy()-41,48y
323 fBodyAcc-bandsEnergy()-49,56y
324 fBodyAcc-bandsEnergy()-57,64y
325 fBodyAcc-bandsEnergy()-1,16y

481 fBodyGyro-bandsEnergy()-49,56y
482 fBodyGyro-bandsEnergy()-57,64y
483 fBodyGyro-bandsEnergy()-1,16y
484 fBodyGyro-bandsEnergy()-17,32y
485 fBodyGyro-bandsEnergy()-33,48y
486 fBodyGyro-bandsEnergy()-49,64y
487 fBodyGyro-bandsEnergy()-1,24y
488 fBodyGyro-bandsEnergy()-25,48y
489 fBodyGyro-bandsEnergy()-1,8z
490 fBodyGyro-bandsEnergy()-9,16z
491 fBodyGyro-bandsEnergy()-17,24z
492 fBodyGyro-bandsEnergy()-25,32z
493 fBodyGyro-bandsEnergy()-33,40z
494 fBodyGyro-bandsEnergy()-41,48z
495 fBodyGyro-bandsEnergy()-49,56z
496 fBodyGyro-bandsEnergy()-57,64z
497 fBodyGyro-bandsEnergy()-1,16z
498 fBodyGyro-bandsEnergy()-17,32z
499 fBodyGyro-bandsEnergy()-33,48z
500 fBodyGyro-bandsEnergy()-49,64z
501 fBodyGyro-bandsEnergy()-1,24z
502 fBodyGyro-bandsEnergy()-25,48z
503 fBodyAccMag-mean()
504 fBodyAccMag-std()
505 fBodyAccMag-mad()
506 fBodyAccMag-max()
507 fBodyAccMag-min()
508 fBodyAccMag-sma()
509 fBodyAccMag-energy()
510 fBodyAccMag-iqr()
511 fBodyAccMag-entropy()
512 fBodyAccMag-maxInds
513 fBodyAccMag-meanFreq()
514 fBodyAccMag-skewness()
515 fBodyAccMag-kurtosis()
516 fBodyBodyAccJerkMag-mean()
517 fBodyBodyAccJerkMag-std()
518 fBodyBodyAccJerkMag-mad()
519 fBodyBodyAccJerkMag-max()
520 fBodyBodyAccJerkMag-min()
521 fBodyBodyAccJerkMag-sma()
522 fBodyBodyAccJerkMag-energy()
523 fBodyBodyAccJerkMag-iqr()
524 fBodyBodyAccJerkMag-entropy()
525 fBodyBodyAccJerkMag-maxInds

Competition Dataset Features

126 tBodyGyro-std()-Z
127 tBodyGyro-mad()-X
128 tBodyGyro-mad()-Y
129 tBodyGyro-mad()-Z
130 tBodyGyro-max()-X
131 tBodyGyro-max()-Y
132 tBodyGyro-max()-Z
133 tBodyGyro-min()-X
134 tBodyGyro-min()-Y
135 tBodyGyro-min()-Z
136 tBodyGyro-sma()
137 tBodyGyro-energy()-X
138 tBodyGyro-energy()-Y
139 tBodyGyro-energy()-Z
140 tBodyGyro-iqr()-X
141 tBodyGyro-iqr()-Y
142 tBodyGyro-iqr()-Z
143 tBodyGyro-entropy()-X
144 tBodyGyro-entropy()-Y
145 tBodyGyro-entropy()-Z
146 tBodyGyro-arCoeff()-X,1
147 tBodyGyro-arCoeff()-X,2
148 tBodyGyro-arCoeff()-X,3
149 tBodyGyro-arCoeff()-X,4
150 tBodyGyro-arCoeff()-Y,1
151 tBodyGyro-arCoeff()-Y,2
152 tBodyGyro-arCoeff()-Y,3
153 tBodyGyro-arCoeff()-Y,4
154 tBodyGyro-arCoeff()-Z,1
155 tBodyGyro-arCoeff()-Z,2
156 tBodyGyro-arCoeff()-Z,3
157 tBodyGyro-arCoeff()-Z,4
158 tBodyGyro-correlation()-X,Y
159 tBodyGyro-correlation()-X,Z
160 tBodyGyro-correlation()-Y,Z
161 tBodyGyroJerk-mean()-X
162 tBodyGyroJerk-mean()-Y
163 tBodyGyroJerk-mean()-Z
164 tBodyGyroJerk-std()-X
165 tBodyGyroJerk-std()-Y
166 tBodyGyroJerk-std()-Z
167 tBodyGyroJerk-mad()-X
168 tBodyGyroJerk-mad()-Y
169 tBodyGyroJerk-mad()-Z
170 tBodyGyroJerk-max()-X

326 fBodyAcc-bandsEnergy()-17,32y
327 fBodyAcc-bandsEnergy()-33,48y
328 fBodyAcc-bandsEnergy()-49,64y
329 fBodyAcc-bandsEnergy()-1,24y
330 fBodyAcc-bandsEnergy()-25,48y
331 fBodyAcc-bandsEnergy()-1,8z
332 fBodyAcc-bandsEnergy()-9,16z
333 fBodyAcc-bandsEnergy()-17,24z
334 fBodyAcc-bandsEnergy()-25,32z
335 fBodyAcc-bandsEnergy()-33,40z
336 fBodyAcc-bandsEnergy()-41,48z
337 fBodyAcc-bandsEnergy()-49,56z
338 fBodyAcc-bandsEnergy()-57,64z
339 fBodyAcc-bandsEnergy()-1,16z
340 fBodyAcc-bandsEnergy()-17,32z
341 fBodyAcc-bandsEnergy()-33,48z
342 fBodyAcc-bandsEnergy()-49,64z
343 fBodyAcc-bandsEnergy()-1,24z
344 fBodyAcc-bandsEnergy()-25,48z
345 fBodyAccJerk-mean()-X
346 fBodyAccJerk-mean()-Y
347 fBodyAccJerk-mean()-Z
348 fBodyAccJerk-std()-X
349 fBodyAccJerk-std()-Y
350 fBodyAccJerk-std()-Z
351 fBodyAccJerk-mad()-X
352 fBodyAccJerk-mad()-Y
353 fBodyAccJerk-mad()-Z
354 fBodyAccJerk-max()-X
355 fBodyAccJerk-max()-Y
356 fBodyAccJerk-max()-Z
357 fBodyAccJerk-min()-X
358 fBodyAccJerk-min()-Y
359 fBodyAccJerk-min()-Z
360 fBodyAccJerk-sma()
361 fBodyAccJerk-energy()-X
362 fBodyAccJerk-energy()-Y
363 fBodyAccJerk-energy()-Z
364 fBodyAccJerk-iqr()-X
365 fBodyAccJerk-iqr()-Y
366 fBodyAccJerk-iqr()-Z
367 fBodyAccJerk-entropy()-X
368 fBodyAccJerk-entropy()-Y
369 fBodyAccJerk-entropy()-Z
370 fBodyAccJerk-maxInds-X

526 fBodyBodyAccJerkMag-meanFreq()
527 fBodyBodyAccJerkMag-skewness()
528 fBodyBodyAccJerkMag-kurtosis()
529 fBodyBodyGyroMag-mean()
530 fBodyBodyGyroMag-std()
531 fBodyBodyGyroMag-mad()
532 fBodyBodyGyroMag-max()
533 fBodyBodyGyroMag-min()
534 fBodyBodyGyroMag-sma()
535 fBodyBodyGyroMag-energy()
536 fBodyBodyGyroMag-iqr()
537 fBodyBodyGyroMag-entropy()
538 fBodyBodyGyroMag-maxInds
539 fBodyBodyGyroMag-meanFreq()
540 fBodyBodyGyroMag-skewness()
541 fBodyBodyGyroMag-kurtosis()
542 fBodyBodyGyroJerkMag-mean()
543 fBodyBodyGyroJerkMag-std()
544 fBodyBodyGyroJerkMag-mad()
545 fBodyBodyGyroJerkMag-max()
546 fBodyBodyGyroJerkMag-min()
547 fBodyBodyGyroJerkMag-sma()
548 fBodyBodyGyroJerkMag-energy()
549 fBodyBodyGyroJerkMag-iqr()
550 fBodyBodyGyroJerkMag-entropy()
551 fBodyBodyGyroJerkMag-maxInds
552 fBodyBodyGyroJerkMag-meanFreq()
553 fBodyBodyGyroJerkMag-skewness()
554 fBodyBodyGyroJerkMag-kurtosis()
555 angle(tBodyAccMean,gravity)
556 angle(tBodyAccJerkMean,gravityMean)
557 angle(tBodyGyroMean,gravityMean)
558 angle(tBodyGyroJerkMean,gravityMean)
559 angle(X,gravityMean)
560 angle(Y,gravityMean)
561 angle(Z,gravityMean)

171 tBodyGyroJerk-max()-Y
172 tBodyGyroJerk-max()-Z
173 tBodyGyroJerk-min()-X
174 tBodyGyroJerk-min()-Y
175 tBodyGyroJerk-min()-Z
176 tBodyGyroJerk-sma()
177 tBodyGyroJerk-energy()-X
178 tBodyGyroJerk-energy()-Y
179 tBodyGyroJerk-energy()-Z
180 tBodyGyroJerk-iqr()-X
181 tBodyGyroJerk-iqr()-Y
182 tBodyGyroJerk-iqr()-Z
183 tBodyGyroJerk-entropy()-X
184 tBodyGyroJerk-entropy()-Y
185 tBodyGyroJerk-entropy()-Z
186 tBodyGyroJerk-arCoeff()-X,1
187 tBodyGyroJerk-arCoeff()-X,2
188 tBodyGyroJerk-arCoeff()-X,3
189 tBodyGyroJerk-arCoeff()-X,4
190 tBodyGyroJerk-arCoeff()-Y,1
191 tBodyGyroJerk-arCoeff()-Y,2
192 tBodyGyroJerk-arCoeff()-Y,3
193 tBodyGyroJerk-arCoeff()-Y,4
194 tBodyGyroJerk-arCoeff()-Z,1
195 tBodyGyroJerk-arCoeff()-Z,2
196 tBodyGyroJerk-arCoeff()-Z,3
197 tBodyGyroJerk-arCoeff()-Z,4
198 tBodyGyroJerk-correlation()-X,Y
199 tBodyGyroJerk-correlation()-X,Z
200 tBodyGyroJerk-correlation()-Y,Z

371 fBodyAccJerk-maxInds-Y
372 fBodyAccJerk-maxInds-Z
373 fBodyAccJerk-meanFreq()-X
374 fBodyAccJerk-meanFreq()-Y
375 fBodyAccJerk-meanFreq()-Z
376 fBodyAccJerk-skewness()-X
377 fBodyAccJerk-kurtosis()-X
378 fBodyAccJerk-skewness()-Y
379 fBodyAccJerk-kurtosis()-Y
380 fBodyAccJerk-skewness()-Z
381 fBodyAccJerk-kurtosis()-Z
382 fBodyAccJerk-bandsEnergy()-1,8x
383 fBodyAccJerk-bandsEnergy()-9,16x
384 fBodyAccJerk-bandsEnergy()-17,24x
385 fBodyAccJerk-bandsEnergy()-25,32x
386 fBodyAccJerk-bandsEnergy()-33,40x
387 fBodyAccJerk-bandsEnergy()-41,48x
388 fBodyAccJerk-bandsEnergy()-49,56x
389 fBodyAccJerk-bandsEnergy()-57,64x
390 fBodyAccJerk-bandsEnergy()-1,16x
391 fBodyAccJerk-bandsEnergy()-17,32x
392 fBodyAccJerk-bandsEnergy()-33,48x
393 fBodyAccJerk-bandsEnergy()-49,64x
394 fBodyAccJerk-bandsEnergy()-1,24x
395 fBodyAccJerk-bandsEnergy()-25,48x
396 fBodyAccJerk-bandsEnergy()-1,8y
397 fBodyAccJerk-bandsEnergy()-9,16y
398 fBodyAccJerk-bandsEnergy()-17,24y
399 fBodyAccJerk-bandsEnergy()-25,32y
400 fBodyAccJerk-bandsEnergy()-33,40y

Annex D

Dataset collection circuit

The circuit (Figure 27) performed in the tests for the dataset collection is represented in the following figure, considering the numbers above each phase as seconds to spend in that activity.

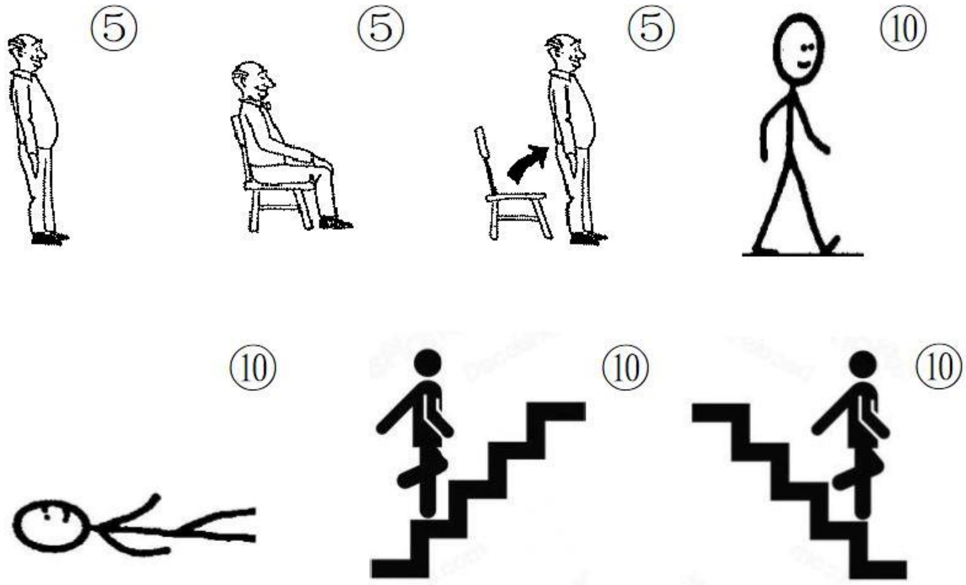


Figure 27: Circuit performed in tests collected with elders.