Universiteit Leiden
Opleiding Informatica

Comparing Sensor Networks

for Activity Recognition

Name: Stylianos Paraschiakos

Date: 28/08/2017

1st supervisor: Arno Knobbe
2nd supervisor: Ricardo Cachucho

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Comparing Sensor Networks for Activity Recognition

Stylianos Paraschiakos

Research Context

This research is part of the requirements for the successful completion of Computer Science master's degree in the Leiden Institute of Advanced Computer Science (LIACS) of Leiden University. The academic supervisors of this thesis are Dr. A. Knobbe with Phd canditate R. Cachucho. This research project took place during a 6 month research internship in the Leiden University Medical Center (LUMC) in the department of Molecular Epidimiology (MOLEPI), in Leiden. The supervisors of the project, were Pr. Dr. E. Slagboom and Dr. M. Beekman.

Preface

I would like to thank all the people who have helped me and contributed to this thesis directly or indirectly.

First of all, I wish to express my sincere gratitude to Ricardo for his daily supervision and guidance. From him, I have learned to be patient and to be critical - two qualities that every researcher must possess. His invaluable comments and suggestions have helped me improve my work greatly. His pleasant attitude has made it enjoyable to collaborate with him.

I am also grateful to my professor, Dr. Arno Knobbe, for giving me the opportunity to work with him. I am thankful for his timely advice, his guidance and constructive feedback that have gone a long way in making this work more worthwhile.

I would also like to thank the members of the Molecular Epidemiology department of Leiden University Medical Center, and especially Dr. Marian Beekman and Pr. Dr. Eline Slagboom, for believing in me and giving me the chance to work with them. You were always very helpful, and I feel honored for trusting me to continue working with you.

Of course, I would like also, to express my special thanks to my friends here in Leiden, for their kind words of encouragement, motivation and support the past two years. You made Leiden feel like home and inspired me for the things coming ahead us.

At last but not least, I want to thank my parents and sister, who are supporting me in all the ways possible all these years and who I own almost all I have achieved until now. I wouldn't make it without you, thank you!

Stylianos Paraschiakos

Contents

Abstract

In recent years, the development of low-cost and energy-efficient sensing technology combined with the evolution of the pervasive computing field open up new opportunities in the human-computer interaction domain. One of the major applications that attracted a wide attention, is human activity recognition, using wearable sensors technology. Although the field has an ever-increasing research interest, there are still a number of challenges on the choices of sensor body location, feature construction and selection, and modeling.

This thesis investigates wearable sensor positioning for monitoring daily living activities of older individuals and investigates different combinations of features and models on different sensor positions. The motivation of this research comes from the medical field of health monitoring and more particularly on the "*Growing Old TOgether*" study of Leiden University Medical Center. The goal is to provide a framework that can answer the following questions, for the given sensor networks and group of activities: (i) Which is the best sensors network? and (ii) Which is the best body location for using only one sensor? Based on that, the analysis pipeline and how to further improve models is discussed.

Chapter 1

Introduction

In this chapter, we will describe the general scope of this thesis. First, the problem of activity recognition will be introduced providing also the motivation of this thesis. Following that, we will discuss the methodology and the different challenges of activity recognition as they are presented in literature. In Section 1.5, the research questions and the contribution of this thesis in *Activity Recognition* domain will be stated. Finally, we will conclude with the outline of the remainder of this thesis.

1.1 Problem Description

In recent years, the development of low-cost and energy-efficient sensing technology combined with the evolution of the pervasive computing[1] field, open up new opportunities in the human-computer interaction domain. Technologies such as fitness monitoring, smart homes, auto driving cars etc. are becoming more and more common.

For those reasons, wearable sensors technology for human daily activity monitoring, has attracted a wide attention. A number of sensors, strategically placed on a human body, can create a network that can monitor physical activities, vital-signs and provide real-time feedback analytics [1].

Sensor-based Activity Recognition

Activity recognition is the task of identifying the actions of individuals via wearable or environmental sensors. Environmental sensors can be cameras, microphones, kinematic sensors and in general sensors located in a room, where the individual takes part in activities. On the other hand, wearable sensors are accelerometers, gyroscopes or sensors, like electrocardiography (ECG), blood pressure, heart rate, breath rate and temperature, located on the body of the individual. The combination of multiple sensors used for the same objective is called a ***Sensor network***.

[1] A concept in software engineering and computer science where computing can occur using any device, in any location, and in any format. Pervasive computing systems are totally connected and consistently available.

In this thesis, we will study human motion based on data of a body-worn sensor network. Human motion can be divided in three types [2]:

- **Actions:** Simple, meaningful motion patterns, which occur periodically or singly.
- **Activities:** A sequence of one or several actions, where those actions can represent the same or different motions.
- **Behaviors:** A chain of one or several activities.

As an example, a footstep is an action, a sequence of footsteps is an activity (walking) and, finally, walking from point A to point B is a chain of activities (standing, walking and then standing again), which is called behavior. The main idea of activity recognition is to link each sensor measurement to an activity label. Therefore, the problem of activity recognition can be defined as follows,

Definition

Given activity labels $A = \{y_0, \ldots, y_{n-1}\}$ *and a series of measurements* $S = [(t_0, m_0), \ldots, (t_k, m_k)]$*, where* t_i *is the timestamp of measurement* m_i*. The purpose of the activity recognition process is to find mapping* $f : m_i \to A$*, so that* $f(m_i)$ *is as similar as possible to the actual activity performed at* t_i [3].

This task is really challenging owing to the complexity and diversity of human activities. To successfully identify activities, Data Mining and Machine Learning learning techniques are used. The physical activity recognition problem can be seen as a supervised classification problem. Several methods have been studied, among others Decision Trees, k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), Naive Bayes, Random Forest, and Neural Networks and Deep Learning [3, 4, 5, 6, 7].

1.2 Motivation

The latest developments of wearable devices have enabled novel applications in different areas, such as healthcare, sports or security, among others, and at this moment, there are many open problems to be tackled [4]. One of the most promising fields is the healthcare domain. Applications like long-term health monitoring, as part of everyday life, is a central element of care in certain areas of health and disease management. Another application lies in the increasing interest of research around healthier lifestyles in elderly population [8, 9]. In many of these studies, accelerometers and other wearable sensors are used for the measurement of the quantity and quality of the physical activities performed [10, 11, 12].

One of these studies, the *Growing Old Together* or GOTO study [10] is the motivation for this thesis. The GOTO study was designed and performed by the departments of Molecular Epidemiology, Gerontology and Geriatrics, Radiology and Medical Statistics of *Leiden University Medical Center* (LUMC) and the Division of

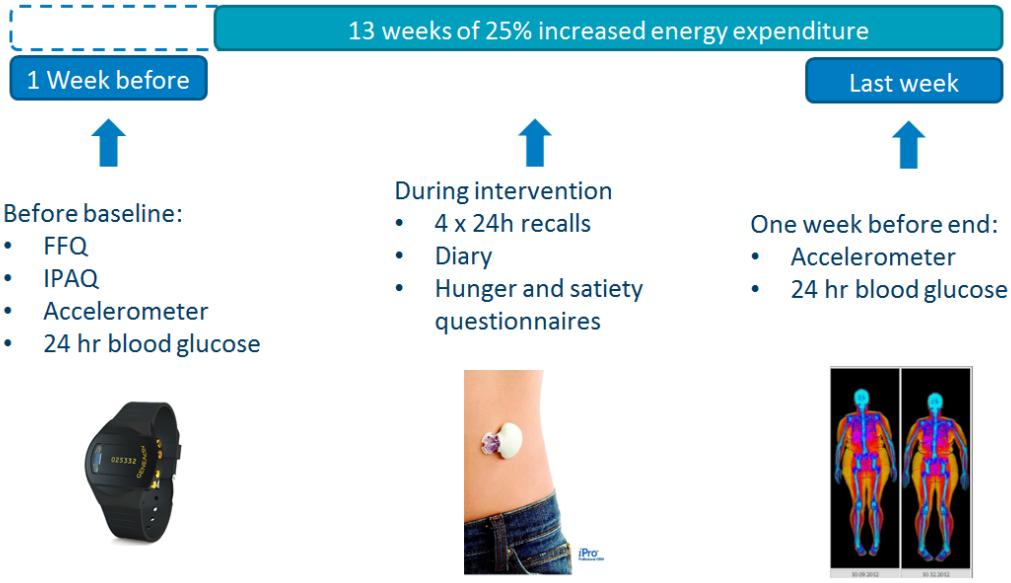

Figure 1.1: The GOTO study[10].

Human Nutrition of *Wageningen University*. The goal of the study was to observe the effects of a 13-weeks lifestyle intervention in 164 older adults (with average age = 63.2 years). In detail, a 13 weeks lifestyle program was applied, with a target of 12.5% caloric restriction and 12.5% increase in energy expenditure through an increase in physical activity. Individual guidelines were prescribed by respectively a dietitian and physiotherapist in consultation with the participant to match the subjects' preferences and physical capabilities. Multiple measurements took place in the beginning, end and during the study (Figure 1.1). Among those, to quantitatively determine physical activity prior to the intervention and at the end of the intervention, wearable accelerometers for seven days, were used on wrist and ankle (GENEActiv, Activinsights, Kimbolton, UK) [10]. However, the arising problem was that except for the accelerometer measurements there was no information about the type of activities performed during those 2 weeks (prior and at the end of the intervention). In order to overcome that problem, there was a need to develop an activity recognition model, that would predict the type of activity from accelerometer data.

Motivated by that, I worked for 6 months as an intern in the department of Molecular Epidemiology in Leiden University Medical Center (March-August 2017) with the goal of developing such an activity recognition model. This thesis reports the research that took part during this period.

1.3 Methodology

In this section, the methodology of solving the problem of activity recognition is discussed. The process to solve this problem can be divided into three main phases: data collection, training, and activity recognition (Figure 1.2).

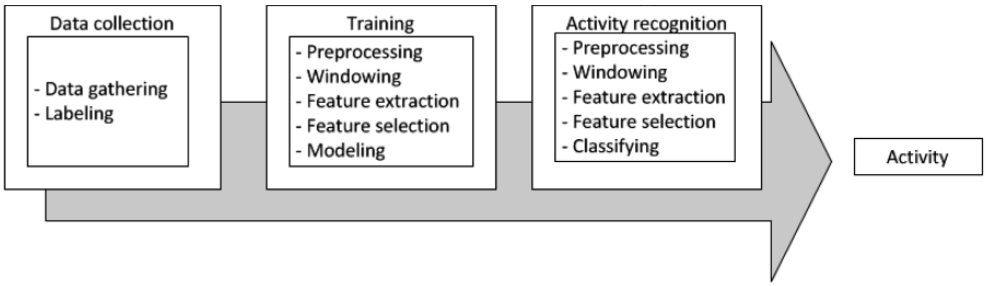

Figure 1.2: The three main phases for activity recognition [5].

1.3.1 Data collection

This is the first phase where the data for model training are collected. If we want to predict n activities, then the training dataset must be collected with at least n activities. Furthermore, the training dataset or labeled dataset should be created using the same devices with the data that we want to predict and placed in the same body parts. A good training data set includes enough variation. This way, the models trained using the data learn exceptional circumstances as well, and therefore, work in the wider spectrum causing less misclassifications [13]. That is because, individuals perform activities in their own way, which creates special patterns in the sensor data.

1.3.2 Training

In this phase, the labeled dataset is used to teach the characteristics of every label (activity) to the model. The learning procedure can be divided in; pre-processing, feature construction and selection, modeling and evaluation.

Pre-processing

It is the stage where the collected data can be modified to make the task of recognition easier. For instance, dealing with missing data, omitting data believed not useful, synchronizing signals, and dealing with different sampling rates. The purpose of pre-processing is to speed-up computation by keeping only the meaningful data.

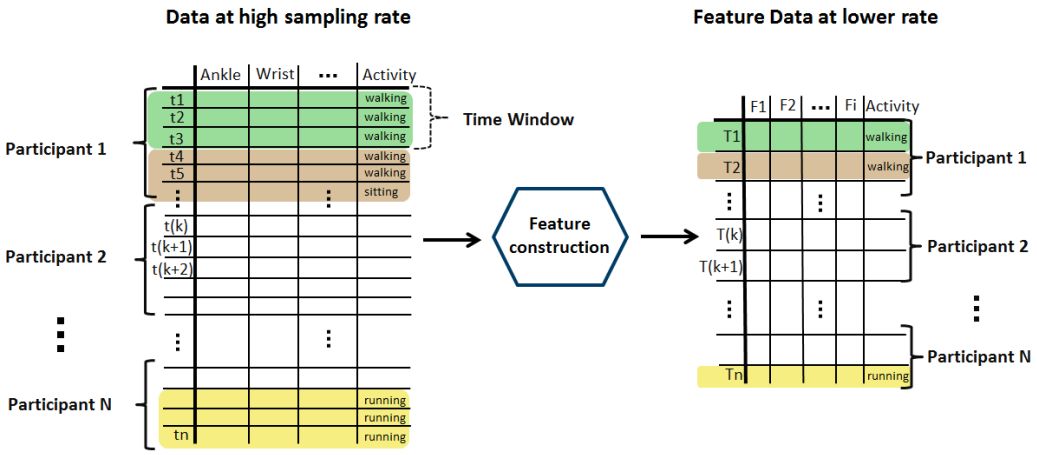

Figure 1.3: The features construction step.

Feature construction and selection

In order to define a label/activity for a certain time period, windowing data is used. This method divides the high rated signal into equally sized windows of time, e.g. 5 seconds. This means that the one does not need to label each sensor measurement separately, but only for every window. For each window, features that represent its measurements are constructed, see Figure 1.3. The values of the feature try to catch the relations of the data in some descriptive way. The most popular features are those constructed by aggregation functions such as mean, standard deviation, maximum, minimum, median etc. [14, 15, 16, 17]. Adding to those, there are also other ways of feature construction such as Fourier Transformations [4], and Entropy calculated over frequency domain coefficients [18].

After feature construction from the signals, selection methods are applied to find the most descriptive ones. At this point, we also have to state that there are some studies which train the methods on raw data without the construction of features [19, 20, 6]. Nevertheless, for this thesis we choose to use this step for the reasons that it will reduce our computational time in training and it is more straightforward to have a time window which represent an activity than just one measurement. For that reason, we compared two methods one with a manually constructed features at a fixed window and one automated (see Section 3.3). Furthermore, the theory behind feature construction and selection is presented in Section 2.3.

Modeling

This is the procedure where classification algorithms are using the selected features as inputs and learn a decision rule or function that associates the input data to the classes. There are main directions in machine learning techniques: *supervised* and *unsupervised* learning. Supervised learning approaches are those that require labeled data and unsupervised that infer automatically the labels from the data.

For our project supervised classification algorithms will be used. Several methods are suggested in the literature for activity recognition [3, 4, 5, 6, 7], among others Decision Trees (DT), Support Vector Machines (SVM), Random Forests (RF), k-Nearest Neighbors (kNN), Artificial Neural Networks (ANN), as well naive Bayes, Hidden Markov Models (HMM) and hierarchical classification. The challenge of choosing the most suitable one will be discussed in Section 1.4.

Evaluation

During this the procedure the models efficiency is validated. Having trained already a model and before apply it, we have to evaluate it using a known test dataset. The most known evaluation method in Machine Learning field is the cross validation [21], where the dataset is split in training and test sets. Then the model is trained on the train set and applied on the test set, which was not used during the training. To evaluate its performance different measures are computed (accuracy, precision). In Section 2.3, we will present this step in detail.

1.3.3 Activity recognition

This is the last phase where the trained model is used to classify the unlabeled data. In order to predict the labels from the unknown data, we have to compute the selected features for the same window size that where used in the training dataset. Those features will be given as an input to our model to predict the activity class. It has to be noted that in order to classify the new data, the same parameters with the train dataset must be used.

1.4 Challenges

In this section, an overview of the challenges that have to be handled during the 3 aforementioned phases is given. There are 4 main choices to be considered before we start training a model for activity recognition.

Type of individuals

The physical characteristics of the subjects, whose sensor data will be used for the model training, play a significant role. Characteristics like the BMI[2] or the age of the individuals taking part in the generation of the training data could change the range and the patterns of activities that the individual perform. This point was studied in [15] where different data collection scenario were tested for different age groups and the resulting data patterns for every group differ. For that reason, since we want to predict activities for older individuals the data we have to use for training have to be from a similar group.

[2]Body Mass Index

Type, number and placement of sensors

It is clear that the task of activity recognition depends on the devices their location on individuals' body. The different type of sensor networks give different kind of inputs, in different sampling rates. This problem it is stated in literature as *The Sensor Selection Problem* [22, 23] and we will discuss it in more detail in Chapter 2. In general, for our research sensor network with both accelerometers and sensors for physical measurements will be used. Moreover, it is pointed in several studies [3, 4, 15, 24, 25] that the sensors placement has a direct effect on the measurement of bodily motions, but the ideal sensor location for particular applications is still a subject of much debate. Various studies combined multiple accelerometers in different body locations, with the majority of highlighting that the placement of many sensors can become burdensome for the individuals [18]. This is leading us to determine both the minimum number of sensors as well as their relevant placement, while still ensuring a sufficiently high activity recognition rate. In this thesis, we will compare different combinations of sensors placements.

Type and number of activities

Those are the classes have to be predicted. As it was stated in Problem Description, Section 1.1, there are different kind of motions, starting from the simple ones, the actions, to a bit more complicated, the activities, and to more complicated combinations the behaviour [3]. The more complicated a motion is the harder to be predicted, since the patterns in the data would be more unique. For example, walking, standing, laying down could be predicted (classified) easier than dishwashing or vacuum-cleaning. Therefore, in order to capture complex activities we need more data, probably from different body locations, otherwise activities that use only one part of the individuals body could not be classified, correctly. For example, dishwashing that uses only the upper body of an individual could be classified as standing, if the individual wears an accelerometer on the ankle. However, classifying dishwashing as standing is not a misclassification, since the individual, actually, is standing. As a result, we can say that dishwashing is standing combined with another activity; washing, as well as ironing is standing and ironing and etc. Those examples refer to the ontological analysis of the activities and how activities are categorized or if there is a hierarchy among them. Several studies have deal with that [26, 27, 28]. In this thesis, we will try to front the challenge of large set of classes with different characteristics.

Machine Learning Method

Concluding, the activity recognition has a high dependency on the method used to build the model. The choice of method lies on which algorithm will be trained on the collected data. As it was already stated, there are several machine learning methods that can be used with their advantages and disadvantages. In the following table (Table 1.1) different activity recognition studies using several methods and their accuracies is displayed.

Table 1.1: Some related work with the number of activities, the sensor networks, and the methods with their accuracies.

Reference	Activities	Sensor Network type & placement		Learning/Accuracy
Bao [32]	20	ACC	wrist, ankle, thigh, elbow, hip	DT(84%), kNN(83%), NB(52%)
Hanai [33]	5	ACC	chest	DT(93.9%)
Parkka [34]	9	ACC/VS	chest, wrist	DT(86%), Hierarchical(82%), ANN(82%)
He [35]	4	ACC	thigh (pocket)	SVM(97.5%)
Khan [16]	15	ACC	chest	ANN(97.9%)
Zhu [36]	12	ACC	wrist, waist	HMM(90%)
Karantonis [37]	6	ACC	waist	DT(91%)
Mathie [38]	6	ACC	waist	DT(87%)
Gjoreski [39]	7	ACC	chest, waist, ankle, thigh	RF(75%-99%)

However, in this thesis our focus is not on the development of a new approach to activity recognition, but rather the selection and placement of sensors in a way to minimize the needed number while maintaining or even improving activity recognition performance. Therefore, we selected the *C4.5 - Decision Tree* (DT) [29] and *Random Forest* (RF) [30] methods for the following reasons. The Decision Tree in one of the most commonly used methods, because of its straightforward classification rules and its computational efficiency (both in training and testing) [14], while it still performs really well in terms of accuracy. The Random Forest was chosen as an improvement of Decision Tree in terms of accuracy [31], since it is an ensemble learning of Decision Trees, which also corrects the decision trees' habit of overfitting to their training set. Adding to that, Decision Trees and Random Forest were also preferred for their ability to implement feature selection procedures (see Section 2.3).

1.5 Research scope and objectives

As it can be interpreted from the literature, there were a lot of studies about which is the best set of choices for number/placement of sensors, type of individuals, features construction and classifiers (see Table 1.1). Every each of those decisions plays a crucial role on the precision of the activity recognition model. Until today there are papers [4] comparing those steps in order to optimize them, with the main debate being around the right sensor body-location and the machine learning methods to be used.

Based on that fact, in this thesis, we will propose a sequence of steps or pipeline, with the goal to maximize the accuracy of an activity recognition model. In each of these step, we will report the improvement that is introduced at the model. In this section, the main objectives of this work and which are the research question to be answered in order to achieve them are defined.

1.5.1 Objectives

We already stated, while discussing the motivation of this work (Section 1.2), that our main objective is to build an activity recognition model, in order to label/classify the activities that took part during the GOTO study [10] as accurate as possible. Therefore, this thesis focus on the search of high-efficient and high-precision data mining techniques, which can be applied to large-scale sensor data for activity recognition. During this thesis we will analyze, apply and compare those techniques to achieve that goal.

Quest of higher accuracy

In detail, we will:

- analyze the role of sensors' placement and number in a sensor network. Adding to that, we will explore, if combining accelerometers and sensors with physical measurements is an advantage,
- investigate if different time windows for construction of features, and the number of features play a significant role in model's accuracy,
- discuss the way of evaluating activity recognition models,
- look into how the number and ontology of activities affects the model's accuracy and try to suggest a way to deal with a large set of classes, using a data-driven activity ontology, and
- suggest methods to improve activity recognition models' accuracy.

1.5.2 Research Questions

Accordingly, to achieve the proposed objectives, we addressed the following research questions:

1. From the given sensor networks, which is the *best sensor network* for activity recognition?
2. Which is the best sensor placement, when using only one of those sensors or *best minimal sensor network*?
3. Is n-fold validation a good measure of evaluation for temporal classification problems?
4. Could some activity classes be combined in order to achieve higher prediction accuracy, without a predefined activity hierarchy?
5. Is post-processing smoothing a good way to improve a model?

1.6 Outline

This work is divided into 6 chapters. In Chapter 2, preliminary theory will be introduced, where sensor selection problem will be defined, while Time Series data and the problem of Temporal Classification will be analyzed. In Chapter 3, the analysis pipeline of this research will be explored, by presenting the steps activity recognition modeling: Data Collection, Pre-processing, Feature Construction and Selection, Training and Evaluation. Following to that, in Chapter 4, firstly, it is investigated how the ontology of classes can improve a constructed model. Secondly, the effects of applying smoothing filters on predictions are explored. Afterwards, in Chapter 5, the results of models built and methods tested are presented. Adding to that, a discussion on the best sensor set-up will take place. Finally, Chapter 6 will contain an overall conclusion with the answers on the research questions and suggested future work.

Chapter 2

Preliminary Theory

In this chapter, an overview of some preliminaries, needed for the rest of this thesis, are provided. We will first define the *Sensor Selection Problem*. Then the concept of *Sensor Data* and their Time Series structure is presented. Finally, we will introduce the problem of *Temporal Classification*, present the classifiers that will be used for modeling and discuss on how to evaluate them.

2.1 Sensor Selection Problem

Collecting measurements from as many as possible sensors, it is believed, that creates a better understating of the subject investigated. However, due to many reasons; energy, location or comfort limitations, the number of sensors should be kept to the minimun. The problem to find the most efficient set of sensors for a given mission is called the sensor selection problem, and it is defined as,

Problem Definition

Given a set of sensors $S = \{S_1, \ldots, S_n\}$, we need to determine the 'best subset' S' of k sensors to satisfy the requirements of one or multiple missions. The 'best subset' is one which achieves the required accuracy of information with respect to a task while meeting the constraints of the sensors.

In an activity recognition system, high classification accuracy is usually desired. This implies the use of a large number of sensors distributed over the body, depending on the activities to detect. At the same time, a wearable system must be unobtrusive and operate during long periods of time. In this thesis, we will test different schemes of sensor networks in order to find the most efficient in terms of accuracy.

2.2 Sensor Data

Sensor data is the output of a device, that detects and responds to environmental inputs. This output can be used to provide information or input to another process. Sensors can be used to detect different physical elements. Some examples are; accelerometers that detect changes in gravitational acceleration or vivo sensors that measure human physiological elements, like heart rate, breath rate etc.

Sampling rate

In the pre-mentioned examples, sensors capture the changes of their subject by sampling in a specific rate. This rate is called *sampling rate* and depends on the device's configuration, which means that it might differ from device to device. The sampling rate is measured in Hz, which is the average number of samples obtained in one second. Those samples compose the sensor data and they are stored in a time order, following the sequence that they were collected.

2.2.1 Time Series

Time ordered data are defined as *Time Series Data* and they are analyzed in a quite different manner from the aggregate statistical approaches that are used in machine learning and other data science activities. That is because, time series values at some point in time, are correlated with past values of the data at some past point in time, or before a certain number of time steps, known as lags. That means that the data are not necessarily independent and not necessarily identically distributed. Therefore, ordering is very important because there is dependency and changing the order could change the meaning of the data. This property is called *Autocorrelation* or serial correlation. Additionally, sensor data between different sensors may also exhibit cross-correlation such that lags in one sensor's measurements may correlate with the values of other sensor measurements.

Concluding, Time Series data are a collection of observations of well-defined data items obtained through repeated measurements over time, like sensor data. More formally, time series of a variable can be defined as,

Definition

Univariate time series $S = (s_t, t \in [1, N])$ *are quantities* $s \in S$ *that represent or trace the values taken by a variable over a period* t*.*

Time Series Autocorrelation

As we already stated time series data are *autocorrelated*. Here we will try to state this formally. *Autocorrelation* means that value y of a time series, at time point or lag $t = i$ depends on n past values:

$$y_{t=i} = \mu + \phi_1 y_{t=i-1} + \phi_2 y_{t=i-2} + ... + \phi_n y_{t=i-n} + \epsilon, \tag{1}$$

where μ the intercept, ϕ the slope coefficients and ϵ the error term or white noise.

2.2.2 Multivariate Time Series

Having data from multiple sensors creates time series taken by several variables. Those time series are named ***Multivariate Time Series*** and they are defined as,

Definition

Multivariate time series $MS = \{(s_{1,t}, s_{2,t}, \ldots, s_{n,t})t \in [1, N]\}$ *are a set of quantities* $\{s_i \in S_i, i = 1, 2, \ldots, n\}$ *that represent or trace the values taken by a set of* n *variables over a period* t*.*

Cross-correlated Multivariate Time Series

If we now have a multivariate time series of m variables that are cross-correlated, for lags of same size for every series, the equation 1 becomes:

$$\begin{pmatrix} y_{1,t=i} \\ y_{2,t=i} \\ \cdots \\ y_{m,t=i} \end{pmatrix} = M + \Phi_1 \begin{pmatrix} y_{1,t=i-1} \\ y_{2,t=i-1} \\ \cdots \\ y_{m,t=i-1} \end{pmatrix} + \Phi_2 \begin{pmatrix} y_{1,t=i-2} \\ y_{2,t=i-2} \\ \cdots \\ y_{m,t=i-2} \end{pmatrix} + \ldots + \Phi_n \begin{pmatrix} y_{1,t=i-n} \\ y_{2,t=i-n} \\ \cdots \\ y_{m,t=i-n} \end{pmatrix} + E \quad (2)$$

where M the intercept set of y's at $t = i$, Φ the slope coefficients and E the error terms or white noise.

Multivariate Time Series with mixed sampling rates

As we discussed, different devices can have different sampling rates, which is also our case. From that, the problem of mining Multivariate Time Series with mixed sampling rates arises. A time series dataset with mixed sampling rates $S = \{s_1, s_2, \ldots, s_p, r\}$ from $p + 1$ variables consists [40] of the following. The *predictors*, which are the first p variables with the same sampling rate. The *response* or *target* r, which is the variable in lower sampling rate. That means that the length (number of data points) of r is shorter than of s, $|r| < |s|$. The sampling rate of the predictors $f_S = q > 1$, which is multiple of the response's sampling rate, $f_S = q \cdot f_r$ (Figure 2.1).

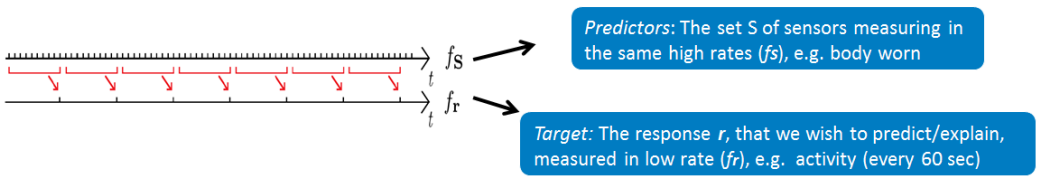

Figure 2.1: Relation between high (f_S) and low (f_r) sampling rates [40].

2.3 Temporal Classification Problem

Mining a sequence of data for machine learning purposes, like time series, is a different task than of learning from static data. Trying to forecast stock prices, monitor patients health or classify activities are all problems with of mining temporal sequences. In this kind of tasks we are looking for a temporal pattern, an episode, hidden in time series that is characteristic and predictive of events that they are more likely to occur frequently. The set of these classification problems are called *Temporal Classification* problems and belong in the more general area of *Sequence Classification* [41].

2.3.1 Definition of the problem

Given L as a set class labels, the task of temporal (sequence) classification is to learn a pattern classifier C, which is a function mapping a sequence s of data points to a class label $l \in L$, written as, $C : s \rightarrow l, l \in L$

Generalizing, the class labels l can be a sequence of classes ls which we want to predict from a sequence of data points, like an activity is a sequence of actions. In other words, the function C will return a sequence of labels ls characterizing the sequence s of data points and not only a single class. This problem is known as *Strong Temporal Classification* [42].

Feature construction and selection

There are three major challenges in sequence classification [41] and in extension in the temporal classification. First, most classifiers, like decision trees and neural networks, can only take input data as a vector of features. However, there are no explicit features in sequence data. Second, eve if we transform a sequence into a set of features, the feature selection is far from trivial. The dimensionality of the feature space for the sequence data can be very high and the computation can be costly. Third, besides accurate classification results, in some applications, we may also want to get an interpretable classifier, which is difficult since there are no explicit features. This challenges can be tackled in 3 ways:

- by feature-based classification, which transforms a sequence into a feature vector and then apply conventional classification methods, like Decision Trees,
- by sequence distance-based classification, where a distance function measures the similarity of sequenced, like Euclidean distance, and
- by a model-based classification, which uses statistical models to classify sequences, like HHM.

In this thesis, we will follow the feature-based classification. To apply feature-based methods on simple time series, usually, before feature selection, time series data needs to be transformed into symbolic sequences through discretization or symbolic

transformation [41]. We already pointed in Section 1.3.2, that the feature can be constructed in different ways, aggregation functions or Fourier transformations. Those feature represent short sequence segments (time windows), which are significantly correlated with one class. Finally, the selection of the most representable features for training our models is taking part.

2.3.2 Classifiers

In this section, we will present the two classifiers selected for the task of activity recognition using as training sets the features constructed and selected. The methods chosen are C4.5 Decision Tree and Random Forests. The reasons behind this choice were discussed in Section 1.4.

Decision Trees

Decision Tree (DT) classification can be thought of as a series of questions, in which the next question depends on the answer to the previous question. After a series of questions, a class label is assigned to the sample. The decision tree decision boundaries in feature space form rectangular decision regions. A decision tree consists of *nodes*, *rules*, *branches* and *leaves*. The first node is called the root node and it performs the first split of dataset into subsets. In each node, a question is asked, which divides the input dataset into subsets, the rule. Depending on the answer, one of the branches is followed to the next node until they reach the leaves, which is the classes, see Figure 2.2. Decision Tree is a top-down architecture. The path from root to leaf node is more of a set of rules; using the rules, we can find the result of classification on the leaf node. To grow a decision tree based on training data, we follow these steps:

1. Start with the feature that best splits the set of items.
2. For the training instances that have been partitioned, continue finding the best feature at each test node.
3. Stop when all training points reached a node that has the same label or when all of the features have been used along the path that reaches the current node.

The method we will use to grow/ learn a Decision tree is C4.5 [29]. This tree classification model, introduced by Ross Quinlan, splits theusing the concept of information entropy. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. When the number of nodes in the tree is reasonable, C4.5 and other classification trees are really power efficient to use, since the chosen rules are interpretable from the user. Therefore, it is used for example in real-time activity recognition applications.

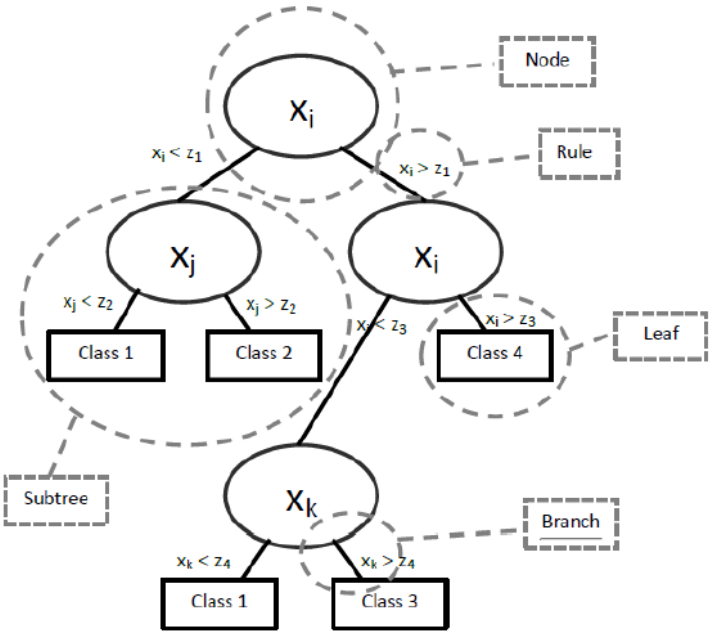

Figure 2.2: A Decision Tree.

Random Forest

Random Forests (RF) [30] consists of an ensemble of decision trees. It improves the classification performance of a single-tree classifier by combining the predictions made by multiple decision trees (bagging), each one generated using a different randomly selected subset of the features. The assignment of a new observation vector to a class is based on a majority vote of the different decisions provided by each tree constituting the forest.

They do not require any domain knowledge or complicated parameter settings and perform very well with high-dimensional data. The main parameters to adjust when using these methods are n estimators (i.e., the number of trees in the forest) and max features (the size of the random subsets of features to consider when splitting a node). Their drawback is that because they are made of several weighted decision trees, they are not easy to interpret [4] (sometimes decisions can be made through voting on contradicting rules). In [43], the authors proposed a classification methodology to recognize, using acceleration data, different classes of motions. They showed that Random Forest algorithm provides the highest average accuracy outperforming the SVMs and the Naive Bayes.

2.3.3 Evaluation

Model Performance

Evaluating the performance of an Activity Recognition model is of high importance. Evaluation is typically conducted using leave-one-out cross validation to assess how the recognition system generalizes to a new situation [7, 44]. To this end, the experimental dataset is partitioned into multiple folds. All folds, except one, are used to train the recognition system. The left-out fold is used for testing. The process is repeated rotating the left-out fold until all folds have been used once for testing. However, while this procedure is similar to the n-folds cross-validation used in machine learning, it has a main difference; folds should **not** be randomly selected from the dataset, since time series data are autocorrelated. As we already stated in Section 2.2.1, a point in time series data depends in the past ones. Adding to that, multivariate time series data from different sensors can also be cross-correlated. For this reason, our training fold should not include any data of the time series of the test set, otherwise we overfit our model. For example, for the task of activity recognition, in order to have a fair prediction, the dataset have to be split in a way that training data do not include data from a participant used in test set. In other words, *Leave-one-participant-out* (LOPO) method has to be used to assess generalization to an unseen user for a user-independent recognition system.

Classification Performance

In LOPO evaluation, we see how our model performed by computing its overall accuracy; proportion of correctly classified examples. However, the accuracy measure does not take into account the imbalanced datasets. In this case, the accuracy is particularly biased to favor the majority classes. Thus, the following measures are suggested [45], confusion matrices' related measures; precision, recall, and F-scores; or graph related like Receiver Operating Characteristic (ROC) curves.

In detail, a *Confusion Matrix* summarizes how many instances of the different activity classes got confused (i.e., misclassified) by the system. Typically, for a binary classification problem, the rows of a confusion matrix show the number of instances in each actual activity class (defined by the ground truth), while the columns show the number of instances for each predicted activity class (given by the classifier's output). Each row of the matrix is filled by comparing all ground truth instances of the corresponding actual class with the class labels predicted by the system, see Table 2.1.

Table 2.1: Confusion Matrix for Binary Classification.

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

Explaining the table, a is the number of correct predictions that an instance is negative (true negatives T_n), b is the number of incorrect predictions that an instance is positive (false positive F_p), c is the number of incorrect of predictions that an instance negative (false negatives F_n), and d is the number of correct predictions that an instance is positive (true positives T_p).

A confusion matrix for non-binary classification has all classes in rows and columns, and the matrix is filled by the number of instances predicted for one class, knowing the actual class. In this way, the main diagonal of the table have the number of instances predicted correctly for every class. An example of activity recognition with 3 classes and 28 instances is given in Table 2.2.

Table 2.2: Confusion Matrix example for 3 classes.

Confusion matrix		Classification result			
		Standing	Walking	Running	Accuracy
True activity	Standing	10	0	0	100,0%
	Walking	0	6	1	85,7%
	Running	0	3	8	72,7%
	Reliability	100,0%	66,7%	88,9%	
				Average Reliability	85,19%
Number of classifications		28			

Average Accuracy	Overall Accuracy
86,15%	
	85,71%

For each activity class the following measures can be computed, using Table 2.1 as a guide:

- *precision*: $\frac{T_p}{T_p+F_p} = \frac{d}{b+d}$
- *recall*: $\frac{T_p}{T_p+F_n} = \frac{d}{c+d}$
- *specificity* $\frac{T_n}{T_n+F_p} = \frac{a}{a+b}$
- *F-measure*: $\frac{2\times precision\times recall}{precision+recall} = \frac{2.2}{2.2+b+c}$

Finally, there are also graph related measures like the *ROC* or *PR* Curves, Receiver Operating Characteristic or Precision-Recall curves, respectively. ROC curves plot the true positive rate (recall) against False-Positive Rate (FPR)($FPR = \frac{F_p}{F_p+T_n}$). It has been suggested that the area beneath the ROC curve can be used as a measure to describe the overall performance of a classifier [46]. It is known as *AUC*, area under the curve and is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.

Chapter 3

Analysis Pipeline

In this chapter, we explain the main steps of the project. We follow the methodology presented in Chapter 1, Data Collection, Training and Activity Recognition. At the beginning, we give; a description of how the sensor dataset was collected and which are the sensor set ups (sensor networks) created. Then, we follow the data pre-processing steps and summarize how the data looked before the stage of training. We continue with the stage of feature construction and selection and the presentation of the methods used for the training. Adding to that, the different models built based on every data set-up and method (features and classifiers) are discussed. Finally, we will close with the evaluation of the models.

3.1 Data Collection

In this section, the phase of data collection is presented. As we already pointed, in order to train a classification model, which will be used to label the unknown data, e.g. GOTO data, we need a training *labeled* dataset. This labeled or annotated dataset has to be created using at least the same set of sensors and labels, with those we want to predict. For that reason, there was a need of a new dataset, where individuals will use the same sensors set with the GOTO study and perform a set of activities that participants of GOTO study expected to perform daily. This data creation procedure was called the *GOTO validation* (GOTOv) study and it will be presented in this section.

3.1.1 GOTOv study

The GOTOv study took place at LUMC, between February and May 2015. During this time, 35 individuals (14 female, 21 male) with an average age of 65 years old took part (see Table 3.1). They wore 6 different devices in different body locations (Table 3.2). The aim of using 6 different devices lies on the fact that the produced annotated datasets would be used in different studies and not only the GOTO, in which only 2 accelerometers (GENEActive) on wrist and ankle were used. Wearing those sensors, every individual performed 16 different activities for around an hour

Table 3.1: The individual and their physiological information.

Code	Age	Gender	Weight (kg)	Height (cm)
GOTOV02	62	female	83	167
GOTOV03	66	male	74	177
GOTOV04	62	female	75	163
GOTOV05	61	female	68	162
GOTOV06	59	male	84	177
GOTOV07	68	male	91	180
GOTOV08	65	male	95	172
GOTOV09	64	male	80	172
GOTOV10	66	male	84	180
GOTOV11	65	male	96	187
GOTOV12	60	male	99	190
GOTOV13	64	female	66	161
GOTOV14	63	male	117	182
GOTOV15	69	male	82	182
GOTOV16	72	female	74	168
GOTOV17	62	female	64	163
GOTOV18	59	male	77	180
GOTOV19	68	female	70	172
GOTOV20	62	male	93	178
GOTOV21	62	male	90	182
GOTOV22	60	male	83	184
GOTOV23	66	female	78	170
GOTOV24	70	female	69	160
GOTOV25	69	male	85	168
GOTOV26	70	female	81	161
GOTOV27	64	male	98	179
GOTOV28	61	female	82	178
GOTOV29	74	male	93	178
GOTOV30	67	male	88	174
GOTOV31	60	female	70	170
GOTOV32	68	female	74	175
GOTOV33	68	male	76	175
GOTOV34	62	male	77	176
GOTOV35	60	male	81	178
GOTOV36	81	female	72	167

Table 3.2: Table with devices and their body locations, used for GOTOv study.

Device	Location
GENEActiv	Right wrist (strap) Chest (belt) Right (strap)
Equivital	Chest* (belt)
Activ8	Upper leg (adhesive tape)
COSMED K4 b2	Nose & mouth (face mask) and torso (belt)
Philips DirectLife activity monitor	Hip (belt) Chest (necklesh)
Polar Electro	Collection unit: attached to K4 b2 belt. Sensor unit: chest* (belt)

and 30 minutes following a specific protocol (see 3.1.3). If a device was severely limiting the participant in his/her movement, it was removed. In detail the devices used:

- The GENEActiv measures tri-axial acceleration (+/- 8g) with a high sampling frequency (88 Hz). It will be attached to the right wrist, right ankle using a strap, and chest with a belt.
- The Equivital belt measures, among others, heart rate and heart rate variability, respiration parameters (vivo measurments) and acceleration (tri-axial). It will be attached to the participant's chest using a belt.
- The Activ8 activity monitor measures acceleration (tri-axial), with built-in activity classification. It will be affixed to the participant's upper leg with surgical tape.
- The Philips DirectLife activity monitor supplies tri-axial acceleration measurements with a sampling frequency of 20 Hz. The activity monitor will be placed around the waist on the hip with a belt and on the chest with a standard necklace.
- The COSMED K4b2 provides information on energy expenditure (indirect calorimetry) by means of a facial mask and sensor unit. The sensor unit quantifies the pulmonary gas exchange (VO2, VCO2) on a breath-by-breath basis. The K4b2 will be attached to the subject's torso using a proprietary belt (equipment weight < 1 kg). The gas exchange analysis will take place by means of a proper mask to be placed in front of subject's nose and mouth. The COSMED K4b2 consists of four devices (see Figure 3.1):

  - Facemask: Plastic hypoallergenic mask, used to collect the expired gases.
  - Interface between the mask and the sensor unit. This consists of a turbine connected to a flow transducer, which is connected to the sensor unit via gas tight tubing.
  - Sensor unit: consists of an O2 as well as a CO2 sensor, a processor and a memory. This unit has also a display and 6 buttons through which the researcher can program and supervise the test. This unit is equipped with an air pump in order to sample the inspired and expired air.
  - Battery: The 6V battery powers the sensor unit for around 2 hours.

- The Polar Electro will provide additional information on heart rate. The device has the size of a watch and will be attached to the participant's chest with a belt. An accompanying small data collection unit will be attached to the K4 b2 sensor unit belt.

Figure 3.1: The devices used in GOTOv study. Active8 and Equivital (right), Philips DirectLife and GENEActive (middle), COSMED K4b2 (left).

Nevertheless, from the aforementioned devices for model training, only a subset was used, the GENEActive accelerometers and the Equivital. The reasons are, firstly, this way we reduce the size of our datasets and as a result the training time for different set-ups. Secondly, from some devices the data were not open available to us or were not useful for activity recognition, but only for energy expenditure models. Finally, in order to built a model for the GOTO study, which is one of our objectives, we needed a dataset from the same sensors that were used in this study (GENEActive ankle, wrist), which was chosen.

3.1.2 Sensor networks

Having concluded on the devices' subset, the following 15 combinations of sensor networks (set-ups) were built and will be used to for training. Our purpose will be

to find the most efficient one. The sensor networks built were:

1. **The 4 minimal ones:**
   GENEActive ankle, GENEActive wrist, GENEActive chest, Equivital
2. **The GENEActive combinations (only accelerometers):**
   GENEActive ankle & wrist, GENEActive ankle & chest, GENEActive chest & wrist, GENEActive ankle & wrist & chest
3. **The GENEActive and Equivital combinations:**
   GENEActive ankle & Equivital, GENEActive wrist & Equivital, GENEActive chest & Equivital, GENEActive ankle & wrist & Equivital, GENEActive ankle & chest & Equivital, GENEActive chest & wrist & Equivital, GENEActive ankle & wrist & chest & Equivital

3.1.3 GOTOv Protocol

It is already stated that the individuals performed 16 activities. However, in order to create the annotated dataset, every individual will have to follow a specific protocol. This way we know for every individual the sequence and the duration of the activities performed. The specific order, duration and description of the activities is presented in Table 3.3. Before every individual starts the sequence of activities, there was a sensor calibration step for the COSMED K4b2, that took approximately 10 to 15 minutes. Following to that, the individuals will synchronize the sensors by lightly jumping up and down for 20 seconds. Subsequently, they will start performing the different activities. The activities performed in two sets. Firstly, indoor activities, resembling those in daily life; lying down, sitting, standing and performing several household chores and secondly, the outdoor activities: walk and cycle, respectively. The second set of activities took part in the immediate vicinity of the LUMC, where individuals encountered ordinary traffic conditions, such as crossroads, traffic signs/lights and different kinds of traffic. We note here that syncJumping was used in order to predict the jumping activity and stepTest as stepping on an object.

While this protocol should ideally be followed by every individual, in a real setup, some sensor data are lost. This was result of some individuals not being able to perform certain activities. Furthermoere, some activities, like the outdoor, could not be performed because of weather conditions. This resulted in some invalid data in the dataset and the actual collected data differ from the expected collection. For that reason, in the end we concluded to a set of 28 out of 35 participants[1], which performed the majority of the activities, while data from all the devices, GENEActive ankle, wrist, chest and Equivital existed. In Table 3.4, the actual time of data inputs for every activity performed is presented. Examining this table it can be seen that there is a class imbalance. Some activities, occurred for prolonged period with respect to others, e.g. cycling versus walking stairs up, which may increase the chance of over-fitting the prediction model [7]. This issue will be analyzed, in Section (3.5).

[1]Participants exlcuded: GOTOV02, GOTOV03, GOTOV04, GOTOV09, GOTOV12, GOTOV19, GOTOV23.

Table 3.3: Activity protocol.

#	**Activity**	**Duration**	**Notes**
1	Sensor synchronization	20 seconds	Participant will lightly jump up and down for 20 seconds to synchronize sensor signals.
2	Standing*	2 minutes	
3	Step test	3 minutes	Participant will step up and down a step 20 times at a pace selected by the participant.
4	Lying down - left	3 minutes	Participant is to turn 90 degrees to the left and remain motionless.
5	Lying down - right	3 minutes	Participant is to turn 180 degrees to the right and remain motionless.
6	Sitting sofa	3 minutes	Participant is to be seated and watch TV, browsing channels occasionally.
7	Sitting couch	3 minutes	Participant is to get seated and read a newspaper.
8	Sitting desk	3 minutes	Participant is to get seated in the office chair and perform some word processing/browsing.
9	Ascending stairs	1 minute	Participant is to ascend a single flight of stairs.
10	Housework dishes	3 minutes	Participant is to wash dishes.
11	Housework stacking shelves	3 minutes	Participant is to stack shelves with books.
12	Housework vacuum cleaning	3 minutes	Participant is to perform some cleaning with a vacuum cleaner.
13	Walking slow pace	5 minutes	Participant is to walk at a slow pace.
14	Walking medium pace	5 minutes	Participant is to walk at a medium pace.
15	Walking fast pace	5 minutes	Participant is to walk at a fast pace.
16	Cycling	15 minutes	Participant is to cycle at a normal pace.

*Between every two activities the participants got some rest by standing, which also provides a clear demarcation between each activity in the signal data.

Table 3.4: Total time, in minutes, of data input for every activity performed.

Activity Label	Minutes
syncJumping	12
standing	66
step	24
lyingDownLeft	104
lyingDownRight	103
sittingSofa	103
sittingCouch	105
sittingChair	101
walkingStairsUp	10
dishwashing	105
stakingShelves	105
vacuumCleaning	106
walkingSlow	124
walkingNormal	124
walkingFast	119
cycling	266

The data collection, were stored in tabular format using; one row for each sample and depending on the data set up, 3 to 32 columns for the attributes. The total size of the sets was estimated to be approximately 53 GB of 26 hours of labeled data per sensor. The format of the different attributes in the data tables were in the following order, depending on the device:

- timestamp, in a date (day/month/year), hour, minutes, seconds, milliseconds format.
- GENEActive ankle attributes: ankleX, ankleY, ankleZ
- GENEActive wrist attributes: wristX, wristY, wristZ
- GENEActive chest attributes: chestX, chestY, chestZ
- Equivital attributes: timestamp, Heart Rate or HR (bpm), Breath rate or BR (rpm), Skin Temperature (IR Thermometer in Celsius), Body Position, Ambulation Status, Device Indications and Alerts, Subject Indications, Low HR Confidence, HR Confidence, Low BR Confidence, BR Confidence, ECG Saturation, Apnea, Heart rate high low, Breathing rate high low, ECG Breathing rate high low, ECG BR (rpm), ECG BR Quality, ECG Lead 1, ECG Lead 2, Lateral Acceleration, Longitudinal Acceleration, Vertical Acceleration, Breathing Wave, Inter Beat Interval (ms)

3.2 Pre-processing

As the data collection performed by simulating a real-world scenario, the raw data contain noise, invalid data, missing values and useless attributes. For this reason, before we conclude to the final training dataset the phase of pre-processing is essential.

3.2.1 Attributes Selection

As our first step in pre-processing, we had to choose which attributes from the devices would be really useful and which not. Omitting the useless ones will reduce the dimensionality of our datasets and following to that computational time. Particularly, we performed a selection from Equivital's 28 attributes (GENEActive has only the tri-axial acceleration). From those attributes, among others we omitted: the ECGs (electrocardiogram), because of high rates (250 Hz) and the ones related with them, and the Alarms and Indications, because they are not appropriate for classification. In the end, we concluded to a set of 12 of them, presented in Table 3.5.

Table 3.5: The included Equivital attributes.

timestamp,	Skin Temperature (IR Thermometer in Celsius),	Lateral Acceleration,
HR-Heart Rate (bpm),	HR Confidence,	Longitudinal Acceleration,
BR-Breath rate (rpm),	BR.Confidence,	Vertical Acceleration,
ECG BR (rpm),	Breathing Wave,	Inter Beat Interval (ms)

3.2.2 Sampling Rates

As pointed out already in 3.1.1, devices sample at different rates. Moreover, one device can sample at a rate depending also in its attributes. At this point we had also to deal with that fact, GENEActive had an 88 Hz sampling rate while the Equivital, depending on the selected attribute the sampling rate ranged between 0.2 Hz - 25 Hz. This differences in Equivital frequencies results in missing values (NA or NaN values). In order to deal with this problem, we decided to use one sampling frequency for all Equivital sets. The frequency used was the accelerometer's one, 25 Hz. The missing values for the ones with lower sampling rates were estimated by interpolation. Furthermore, while combined the datasets for the GENEActive - Equivital combination (data set-up), we needed to downsample GENEActive's 88 Hz sampling rate to 25 Hz by averaging almost every 4 ($88/25 \approx 3.5$) of its inputs.

3.2.3 Timestamps

The timestamp attribute is essential in order to be sure that all devices are synchronized and in order to label the raw data according to the protocol. Therefore, we need to convert all timestamps to a single format, since every device uses different formats. The format we choose was the UNIX time format (in milliseconds), also known as

POSIX time or epoch time. It is a system for describing instants in time, defined as the number of seconds that have elapsed since 00:00:00, Thursday, 1 January 1970, Coordinated Universal Time (UTC) minus the number of leap[2] seconds that have taken place since then [47].

3.2.4 Synchronization

Having the same time format for all the devices, and in order to synchronize (sync) them, we compared the signals by plotting the accelerometers' *Sum Vector Magnitude* (SVM)[3] for both devices. Following to that, we synchronized the devices by checking the local times of the collected data in protocol and timezones that devices were using, an example can be seen in Figure 3.2.

3.2.5 Labels

Since all data set-ups were now synced and in the same sampling rates, we are able to add the activity labels following the data collection protocol see Figure 3.2, right. Moreover, as it is visible in the right figure, between two activities, the dataset contains unlabeled signals. Those signals, are the signal of the transition phase from one activity to the other and do not correspond to any activity. For that reason, they are treated as noise and removed from the sets used for training.

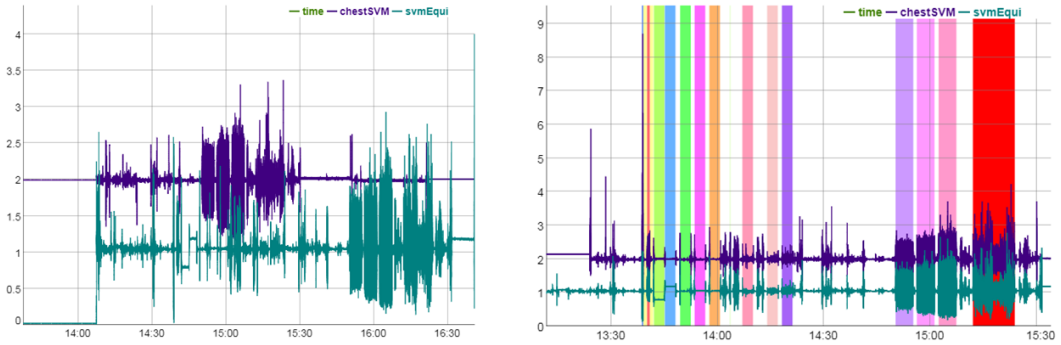

Figure 3.2: Left: Unsynchronized devices (difference of an hour) and Right: synchronized signals. The purple is the SVM signal from GENEActive and the blue-green of the Equivital.The colour shadings represent the different activities according to the protocol.

[2]Leap seconds are scheduled by the International Earth Rotation and Reference Systems Service and are not predictable.

[3]Sum Vector Magnitude or SVM at a t_i moment, SVM $= \sqrt{AccX_i^2 + AccY_i^2 + AccZ_i^2}$.

3.2.6 Data Description

In the end for every data set-up, we merge their datasets accordingly and bind the data of all 28 participants. The datasets used where without any missing or transition data. The concluded training datasets and their attributes have the following order:

1. timestamp, in UNIX milliseconds format
2. GENEActive ankle attributes, 3 to 9 columns
3. Equivital attributes, 13 columns
4. activity label

3.3 Feature Construction and Selection

At this point, our datasets are in time domain, since each sample captures values from different sensors at a certain timestamp. Those datasets can be used for training machine learning models, however transforming them in a way that captures the temporal information we could achieve higher prediction accuracies. In order to achieve this temporal representation, we should transform the raw datasets to more meaningful ones by feature construction and selection. In Sections 1.3 & 2.3, we discussed in detail the importance of windowing our data and constructing suitable features representing those time windows.

For our implementation, we compared three scenarios using aggregated functions for feature construction:

- The Baseline scenario, with only one feature; the mean value, and a fixed window of 1 second was used.
- A fixed window of 1 second again, but with 5 features; the mean, median, standard deviation, minimum and maximum.
- The use of *Accordion* algorithm [40], which given a max window, 5 second for us, constructs and selects features from aggregated functions, on sliding windows equal or smaller to the max one, in an automated way. The set of aggregated functions used are almost the same with the second scenario. In next Section (3.3.1), Accordion will be presented in more detail.

The choice of window sizes, from 1 to max 5 (for Accordion) seconds, was taken after discussion with the LUMC researchers and respectively to their research goals for the use of the constructed model. Furthermore, is one reasonable choice classifying our every day activities set, since for some activities, like walking stairs up, lasted less than 20 seconds. Therefore, having larger windows may conclude in high misclassification rates, depending on the really small training set. Nevertheless, in literature the windows used range, mainly, between 0.5 to 10 seconds, depending on the understanding of the particular problem. In general, longer windows will

be richer in terms of information, and smaller ones will have the ability to reflect more quickly the classes' differences. Summarizing, there will always be a trade-off between quality of features produced and ability to recognize changes in terms of the classification.

Concluding, there is no need of feature selection in baseline scenario (only one) and accordion scenario (automated selection). For the second scenario with five features per attribute, we decided to select them all, since it is a small set. Adding to that, training a C4.5 Decision Tree method integrate, already, a kind of feature selection using the concept of information gain (see Section 2.3.2).

3.3.1 Accordion

The feature construction and selection step is really important in order to represent and summarize the space of training data to something more meaningful. The choices of window sizes, static or sliding and overlapping, the decisions of functions to aggregate those windows, time-domain or frequency domain, and selection of the most representable, manually or with another method, is a backbone of many activity recognition projects. Therefore, we tried to minimize the decisions taken with the use of *Accordion* [40], a method that its authors argue to solve these issues in an automated and memory-conscious approach of feature construction and selection. In more detail, Accordion is an iterative procedure which dynamically constructs aggregated feature candidates and in every iteration evaluates them for selection.

The algorithm was built for mining multivariate time series with mixed sampling rates, by aggregating raw datasets in high sampling rates (predictors) to one with lower (target), as ti was presented in paragraph 2.2.2. The method takes as an input the predictor's sampling rates and the size of the window, e.g. 5 seconds, and aggregates the measurements using a set of 7 functions, as presented in Table 3.6.

Table 3.6: The set of Accordion's aggregate functions.

(1)	the mean value: $avg = \frac{1}{n}\sum_{i=1}^{n} a_i$
(2)	the median
(3)	the maximum value, max
(4)	the minimum value, min
(5)	the standard deviation: $stdv = \sqrt{\frac{\sum_{i=1}^{n}(a_i-\overline{a})^2}{n-1}}$
(6)	the inter-quartile range: $IQR = inf\{x \in R : 0.75 \leq P(X \leq x)\} - inf\{x \in R : 0.25 \leq P(X \leq x)\}$
(7)	the root mean squared: $RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2}$

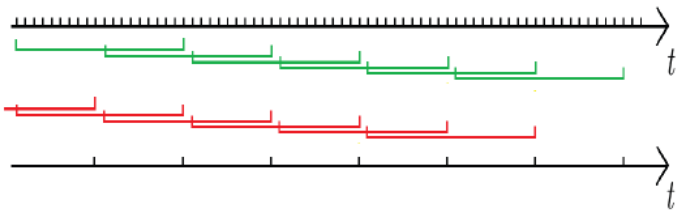

Figure 3.3: Sliding windows.

Using scoring functions (information gain [29]) and heuristic methods the algorithm performs a grid search over the available predictor time series S and the set of aggregate functions A, for every sliding window with size $w \leq w_{max}$ (see Figure 3.3). Then, it returns the set of aggregated features F that gives the best representation of the predictors space, where every feature $f \in F$ is characterized by the aggregation function $a \in A$ and its window size w_{best}. For example, for a given max window size $w_{max} = 5$ seconds and a sampling rate of 20 Hz ($5 \cdot 20 = 100$ inputs), Accordion will slide windows of sizes from 1 to 100 inputs, every 100 inputs, and in each window it will construct the features using the aforementioned functions. Then, for every window size it will evaluate the constructed features (with information gain function) and it will select the best ones. In order not to do a greed search, e.g. all windows combinations with sizes 1 to 100 inputs, for every 100 inputs of the dataset, which will extend the computational time, Accordion gives you the option to select the samples taken for every max window. For our implementation, we used Accordion with a max window size of $w_{max} = 5$ seconds and 10 samples per window, which means the feature were constructed for the sliding windows with sizes $\frac{1}{10}w_{max}, \frac{2}{10}w_{max}, \ldots, w_{max}$. Using the set of the selected features we can train our activity recognition model.

3.4 Training

The last stage, before activity recognition, involves the training of a classification model able to predict the every day tasks performed by the individuals. Our objective is to compare the different models trained on the different sensor networks and obtain the one that maximizes accuracy. The methods to train the combinations of 15 data set ups, with 3 feature construction and selection scenarios were, as it was already stated, two: the C4.5 Decision Tree and the Random Forest, see 2.3.

During this stage, the parameters for those methods were chosen. We did not an extensive search on which parameters would be the most efficient, since our goal is to compare the different sensor networks. For the C4.5 Decision Tree, we used the implementation of J48 in the *rWeka* package in *R*. The main parameter to choose is the minimum number of instances per leaf. For that we kept the default one, which is 2, since it is suggested to work well for most datasets. For the Random Forest, the main parameters to tune are the number of variables randomly sampled

as candidates at each split and the number of trees to grow. Here we kept the number of trees grown the same for all set ups and equal to the default value, 500 trees. For the numbers of variables chosen at each split, we used the suggested $\sqrt{n}$, where n is the number of features for every set up. It is suggested that tuning correctly those parameters can significantly increase the accuracy of the Random Forest model. However, it is a step which needs a lot of testing, and since we had 15 different data set-ups we avoided it and used the suggested ones. Since we decided for the parameters, we trained the two methods on the 3 different sets of features for every data set-up.

3.5 Evaluation

Having now the models trained for every combination of sensor and their features, we need to compare their performance. In order to do that we use leave-one-participant-out (LOPO) cross-validation, as it was described in Evaluation paragraph of 2.3. This evaluation methodology splits the dataset into N folds, where N is the number of subjects (in this case N = 28), using N - 1 folds as the training set and the remaining fold as the test set. Through this procedure, we guarantee that the classifier learns nothing about the subject to be predicted, thus decreasing learning bias and achieving accurate results for cross-person prediction. Note that an evaluation methodology that does not use LOPO would provide better results in terms of accuracy because the model would have already encountered instances from the target subject, but its results would not be representative of the actual prediction capabilities during cross-person prediction or during a cold start [44]. In order to prove this point, we compared different schemes of evaluation using the regular n-folds cross validation versus n-folds cross validation but with folds **not** randomly selected with different sizes of folds. In the latter one, we used also the LOPO one with the folds split depending on the subjects.

Furthermore, having proved our point, we evaluate the classification using a *Confusion Matrix* and its metrics, as presented in Evaluation paragraph of 2.3. As we pointed out in 3.1.3, our classes are imbalanced (see Figure 3.4). For that reason, those metrics are really important to understand our results.

Concluding, with the evaluation results we can confidently answer to the question of the best sensor network. Nevertheless, after the choice of the network and its activity recognition model, there are more steps in order to improve this model. Those steps will be discussed in the next chapter, Chapter 4.

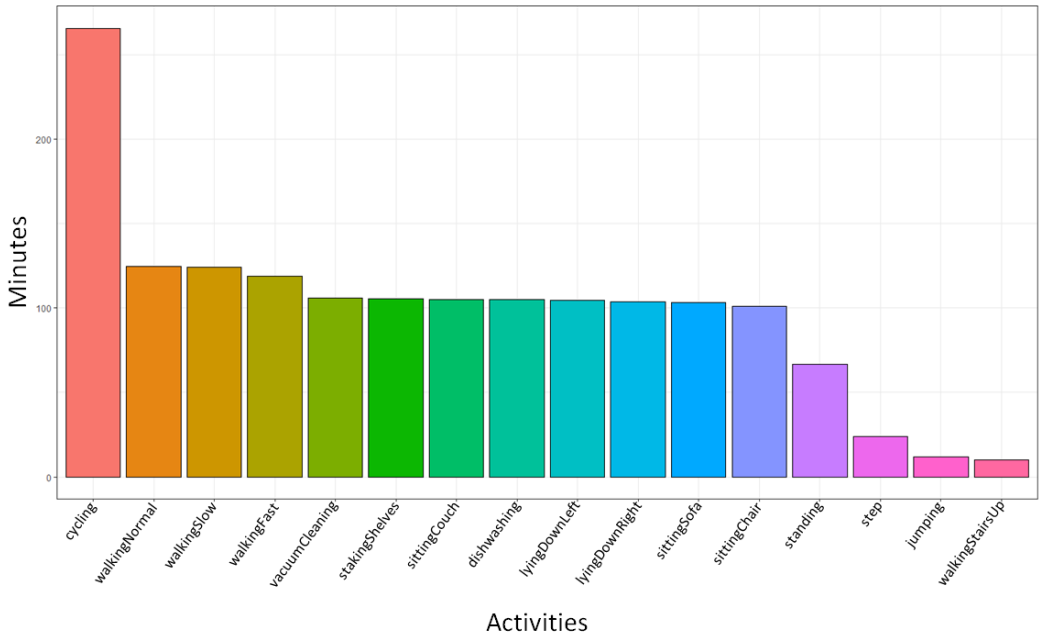

Figure 3.4: Minutes of activities performed from the 28 participants.

Chapter 4

Improving Accuracies for Activity Recognition

In this chapter, we discuss two post-processing analysis steps which optimize an activity recognition model after its training. This is achieved, firstly, by investigating the nature of activities and how they are blended with each other. In order to do that, we will discuss about activity ontology and we will suggest a data driven one. Secondly, this chapter concludes by examining the effects of smoothing in a model's accuracy and suggests a smoothing based on the temporal nature of activity recognition models.

4.1 Dealing with large sets of classes

The different classes of walking, sitting, lying and households, construct groups of activities that have similar patterns. As a result, those similarities are harder to understand and be predicted by a classification model. The similarities of those activities lying on the fact that they belong in a broader family of activities, e.g. walking slow, normal, fast belong all in the family of walking, or sitting in a sofa, couch or chair in the family of sitting. Both of those families have a basic pattern which may confuse one or another sensor depending on its body location e.g dishwashing missclassification to standing from a sensor located on ankle. Therefore, having a predefined activity ontology can improve the modeling phase, since activity patterns can be classified from the simplest ones to the more complex following the ontology tree [26, 27, 28].

4.1.1 Pre-existing tree ontology

For those reasons, it is essential to illuminate the hierarchy of each activity. By that, we mean to investigate how the complex activities are formulated and in which simpler group they belong. In Chapter 1, we defined activities as a sequence of actions, which represent motions. Having this definition in mind, one can divide the activities by the different motions of upper or lower body. However, this is not the

only way to group activities. Other examples could be, grouping activities by their intensity to passive or active, or by the location they took place, indoor or outdoor. Notwithstanding, even having a predefined ontology, sorting activities is not always clear, see Figure 4.1. In this figure, categorizing dishwashing as *Passive* or *Active* activity is not clear.

As it is observed, there is not one way to distinguish the activity hierarchy. In order to design such a hierarchy it is often needed a domain knowledge. However, the ontology is designed, specifically, for each user case. Subsequently, a predefined activity ontology can be challenging.

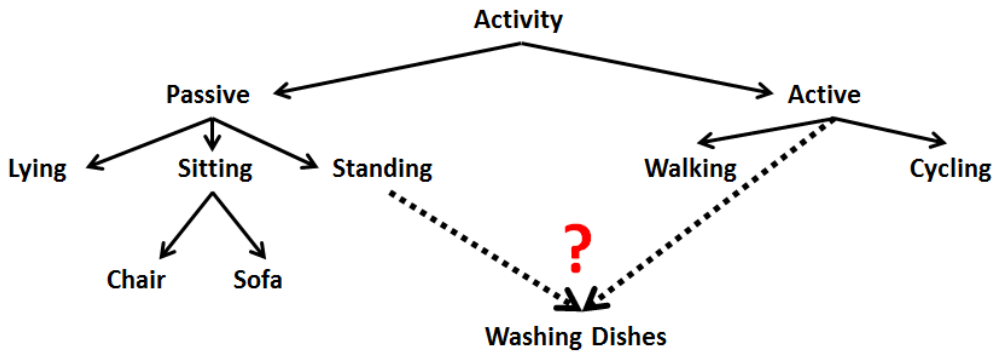

Figure 4.1: Tree Ontology example.

4.1.2 Data Driven tree Ontology

With the purpose of simplifying this step, we would like to introduce a method which suggests an activity ontology depending on the sensors signals. This data-driven ontology is based on the fact that a model is already built.

The method introduces a construction of an activity hierarchy tree, taking advantage of the activities confusion matrix. The tree is grown in a bottom up approach, similar to *Agglomerative Hierarchical clustering* [48].

In detail, in every step, the method collapses two activity classes to one, depending on the amount of their confusion. In this way, the two classes and the merged one create a subtree where two branches from those activities are connected with a parent node class, the merged one. The classes chosen, in every step, are the ones with the higher confusion in the matrix. After the collapsing, the new merged class will replace the previous two and the new confusion matrix will be computed. Subsequently, a tree is built combing every time two classes, child nodes, to one parent node, until only two parent classes are left. So, for a set of n activities the method will do $n-1$ merges. The constructed ontology tree, can be used as an advisory map to build the activity ontology tree.

Summarizing the steps:

1. Compute models confusion matrix.
2. Find max confusion point (max element out of main diagonal) and replace them by the merged one.
3. Collapse classes which give the max Confusion to one class
4. Recompute new_Confusion matrix
5. Iterate from 1 to 4 until two classes are left

Furthermore, the algorithm in every step reports the accuracy gained, as the confusion is getting reduced step by step (tends to zero in the last steps). This way, the user can have a better understanding of the classes to be predicted and if by merging a group of them the task of activity recognition can be more clear. For example, classifying one class of walking instead of multiple classes, might be a better choice, since every walking pace can really differ for every person. As a result, combining the walking paces to one walking the user can be sure that they are classified correctly, and based on that try to understand the different walking patterns by grouping individuals or by using measures like, energy expenditure or activity intensity.

Finally, we have to note here that if the set of classes is imbalanced, the max confusion in the their matrix could be different of the one of a matrix with balanced ones. For example, in a set of 3000 inputs with 4 classes, two of which, A and B, with 1000 and 100 inputs respectively, are confused as following:

- A: by 150 out of 1000 with a class C
- B: by 40 out of 100 with a class D

The max confusion chosen from the confusion matrix would be the one of A, with 150. However, if we examine it more carefully, using the precision measure, for A only the 15% of classes are confused, while for B the 40%. As a result, there two ways to deal with this example; keep the matrix as it is and always merge the classes with the most inputs confused, or transform the matrix to one with the precision numbers per class and then merge. For the first scenario, the merging will be more greedy, which will give the user the view of which classes can be merged to gain faster higher model accuracy. On the other hand, the second scenario, could give a better understanding of which activities have similar signals in the dataset. In Section 5.6, this data-driven ontology is tested.

4.2 Temporal Nearest Neighborhood Approach

In this section, we will examine if applying smoothing filters to a predicted set can improve the overall accuracy of a model. Firstly, *smoothing* in general, is the procedure that modifies individual data points in order to reduce noise. In classification, smoothing can be the process in which predicted data that do not follow the general trend are considered noise and they are being modified.

In Figure 4.2, an example of a data point not following the trend is demonstrated. In this example, in a neighbourhood of five predicted data points, there is one which differ from the other four. This contrast is usually caused by missclassification, since these five activity classes could represent a window of five seconds, with one prediction per second. As a result, classifying an input as sit between walking is probably a false class (noise). Smoothing this window of five second would change the class of sitting to one of walking. Similarly, we can apply a smoothing filter among all our predictions altering classes that are considered noise.

This can be achieved by voting; for every data point, located in the middle of a window, we check if it is equal with the majority of the window (e.g. 5), and if not it is altered to that. Afterwards, we slide the window to the next point and repeat the voting, until all data points are checked. This way we can increase the overall accuracy of our predictions.

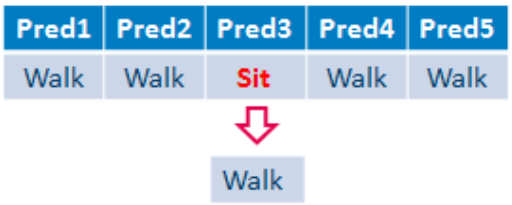

Figure 4.2: Example of noise in predicted dataset.

4.2.1 Temporal Nearest Neighbour Smoothing

Nevertheless, we have to choose wisely the window of data points, in which the filter is applied. In temporal data, and especially in activity recognition the size of window depends on the type of activities, and it is similar with the choice of window for feature construction. For example, if in Figure 4.2 the predictions are per minute and not per second, changing sitting to walking could be incorrect, since an individual could sit for a minute to rest. Therefore, deciding on the window can be done by understanding the nature of the dataset.

Another example can be seen in Figure 4.3, for a window of 5 seconds the sit in the middle will become walk. However, for a window of 3 seconds, starting from Prediction 1, the Prediction 2 will be changed to sit and then, when the sliding window goes to Prediction 2, 3 and 4, no change will take place, since now sit are the majority in the window and also no other changes will take place in the next sliding windows.

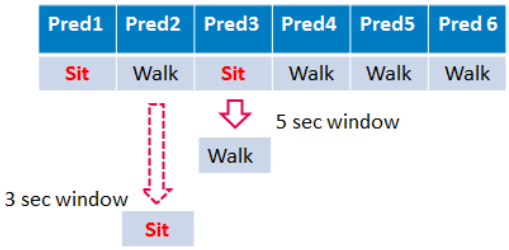

Figure 4.3: Example of noise in predicted dataset.

In general, the temporal data are correlated with the ones in their surrounding neighborhood. As neighborhood of a data point, we can define the number of data points which we expect to have a similar behaviour. Having that in mind we can use the voting system for different sizes of neighborhoods for our model and evaluate it. This way we can search for the optimal size.

4.2.2 Multiple Temporal Nearest Neighbour Smoothing

Furthermore, it is possible to apply multiple smoothing filters with different windows one after the other. However, the choice of order that the filters are applied can produce different results. For example, in Figure 4.4, if we first apply a 3 second window filter and then a 5 second the first filter won't do any changes and the second will change only the Prediction 3 form sit to walk. On the other hand, if the 5 second filter is applied first and then the 3 second one, both sit predictions will alter to walk. Moreover, the same filter can be applied more than once, before applying another one.

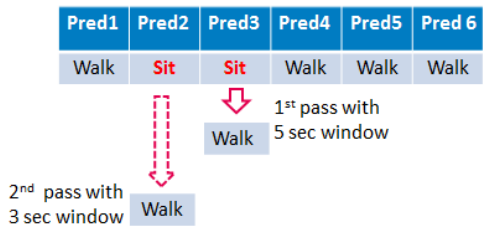

Figure 4.4: Example of noise in predicted dataset.

In Section 5.7, both single and multiple temporal nearest neighbor smoothing approaches are tested and compared with each other.

Chapter 5

Experimental Results and Discussion

Our main goal in this chapter will be to introduce the steps and decisions taken, in order to achieve our objectives for the quest of the higher accuracy (as presented in 1). We will compare all the different strategies and their influence to the models performance and conclude to the most efficient combination of steps. First, we will start with a comparison of n-folds cross-validation and LOPO, in order to prove the point from theory that regular cross validation is not suitable for evaluation of activity recognition models. Then, the different ways of feature construction and selection will be presented and we will continue in the training of the models. Afterwards, the main table of results will be introduced and an exploration about the best sensor network will take place. Subsequently, we will investigate the accuracies of activities per sensors' set-up and a review on a data-driven activity ontology will be given, as suggested in Chapter 5. Finally, we will close the chapter by comparing the 2 temporal Nearest Neighborhood smoothing methods.

The experiments conducted using *R* [49] and the libraries *rWeka*, for C4.5 Decision Tree, *randomForest* for the Random Forest, and *caret* for the evaluation part [50, 51, 52].

5.1 LOPO vs Cross Validation

This section, presents the evaluation methods used to obtain the results. The goal of the evaluation is to estimate the behaviour of the classifiers, in everyday life scenarios, in order to simulate how they would perform in named situations. The commonly used standard n-fold cross-validation (CV) is not adequate for this task [44]. In Section Chapter 2, we already argued why n-fold cross validation is over-optimistic and we presented the *Leave one participant out* or LOPO evaluation. Nevertheless, to prove our point we applied both techniques and compared them.

We experimented it by using different folds and folds selections. In Table 5.1, the results of those experiments are presented. On the left part, the regular n-folds CV is displayed, where for different n the Decision Tree was evaluated (for different data

set-ups). In this part, in order to have random selection of inputs for every fold, we permutated the dataset before we divide it into equal segments. Then we trained the model in the $n-1$ folds and tested it in the one left out, for all n-folds and reported their median accuracy. On the right part of the table, the reported results are the outcome of dividing similarly the datasets in n equal segments (except LOPO), but without permutating the dataset before. As a result, since the datasets were sorted by participants (see Figure 5.1). Therefore, depending on the segments size, some parts or complete participants were not in the train set. Adding to that, for the LOPO evaluation every one segment, out of the 28, is the data of one participant.

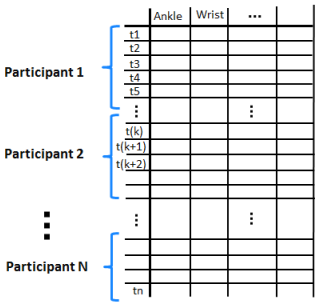

Figure 5.1: The view of the datasets structure.

The outcome of this procedure was that the permutated folds accuracies were $\approx 20-30\%$ higher than for the non-permutated, and also size independent. On the other hand, for the non-permutated it is clear that for the segments that miss complete or big parts of participants in training, the accuracies are lower than from those having at least a part of all of them, like the 1000 one. This significant performance difference was expected, since participants independent evaluation, does not take in account the dependency of participants time-series. Therefore, it leads to highly optimistic results as observed in Table 5.1. Having this point cleared, we used LOPO evaluation for all the models.

Table 5.1: Accuracy results for the two CV evaluations, with different number of folds. On the left the folds were randomly selected (Permutated), while on the right not (NonPermutated).

		Accordion (DT) nFold validation											
		Permutated						NonPermutated					
	DataSetUps	*10*	*28*	*100*	*300*	*500*	*1000*	*10*	*LOPO*	*100*	*300*	*500*	*1000*
GA	ankle	86,9	87,1	87,1	87,2	86,9	87,1	64,1	64,6	66,4	70,9	76,6	80,3
	wrist	86,7	87,0	86,9	87,0	87,0	86,9	63,0	63,8	65,0	68,9	74,2	78,4
	chest	86,4	86,8	86,9	86,8	86,8	86,8	54,6	54,5	60,4	66,7	71,7	76,4
	ankle_wrist	94,4	94,5	94,5	94,5	94,6	94,6	75,4	77,3	78,4	82,3	87,1	89,6
	ankle_chest	92,9	93,0	93,2	93,1	93,2	93,1	68,7	70,0	73,5	80,3	84,3	87,1
	wrist_chest	92,8	93,1	93,2	93,1	93,2	93,1	69,7	71,9	73,0	80,8	84,4	87,3
	ankle_wrist_chest	92,7	92,8	94,5	92,7	92,7	92,7	68,4	67,9	69,5	78,1	83,9	86,4
EQ	equivital	93,8	94,0	94,0	94,2	94,2	94,2	56,9	58,2	63,0	69,5	74,9	80,8

5.2 Feature Construction and Selection

In this section, a description of the constructed features will be given. As it was discussed already, we applied 3 different approaches for feature construction; 1 feature/fixed window 1 sec, 5 features/fixed window 1 sec and Accordion [40]. To evaluate them we trained a Decision Tree, using the data set-ups of *GENEActive*. Their performance results are demonstrated in Table 5.3 and were outcome of the LOPO evaluation. In this table, it can be seen that the Accordion approach outperforms the two others, as it was expected.

In detail, the 1-feature approach, using only the mean value to represent the 88 inputs per seconds (GENEActive's sampling rate = 88 Hz), had poor performance since its accuracy was between $\approx 31\%$ to 64%. For the 5-feature approach, it is clear that adding more elements representing the space of the datasets, has a big effect in terms of accuracy. The 5-features Decision Tree had an increase of more than 10% comparing to the Baseline's accuracy, and in some set-ups (chest, ankle_chest, wrist_chest) more than 20%.

Table 5.2: Number of features created by Accordion for every sensors set-up.

Data Set-Up	Number of features
ankle	140
wrist	224
chest	164
ankle_wrist	148
ankle_chest	119
wrist_chest	163
ankle_wrist_chest	118
equivital	186
ankle_equi	64
wrist_equi	163
chest_equi	138
ankle_wrist_equi	85
ankle_chest_equi	96
wrist_chest_equi	143
ankle_wrist_chest_equi	222

Finally, we ran the Accordion experiments, where different number of features were constructed depending on the dataset. In Table 5.2, the number of feature constructed per set-up is demonstrated. When we introduced Accordion, we described the way it constructs and selects features for sliding windows ≤ 5 seconds (5 times the sampling rate ≈ 450 inputs). In Figures 5.2 & 5.3, some examples of the constructed features are presented. The feature name is denotes the attribute summarized by the function, the second part, and the number of inputs, third part, e.g. in Figure

Table 5.3: Comparing feature construction approaches for GENEActive.

Data Set-Ups	Features		
	1	**5**	**Accordion**
ankle	50.3	62.9	66.0
wrist	43.3	56.1	63.1
chest	31.4	50.8	57.4
ankle_wrist	64.3	73.5	77.8
ankle_chest	52.5	68.4	71.5
wrist_chest	42.0	67.7	74.3
ankle_wrist_chest	51.0	70.0	71.8

5.2, the first feature, ankleY_SD_39, is the standard deviation of ankleY (g force on y-axis) summarized for a window of 39 inputs.

The resulting Decision Trees outperformed the 5-features approach with a range of ≈ 3 to 5%. Nevertheless, since the results where similar we applied a Kolmogorov-Smirnov test to compare their distributions and it was proved that those differences are significant, since p-value was < 0.05. For that reason, we concluded in the use of Accordion's features, in order to train our models. In Figure 5.4, the evolution of the performance in every data set-up versus feature strategy is presented. It is clear that sing Accordion we increased our models'accuracy.

```
"ankleY_SD_39"                                  "ankleY_RMS_31"
"ankleY_Mean_128"                               "Longitudinal.Acc_SD_107"
"Vertical.Acc_Mean_40"                          "ankleZ_RMS_128"
"Vertical.Acc_SD_120"                           "ankleX_SD_128"
"ankleX_RMS_100"                                "ankleZ_Max_108"
"ankleY_RMS_5"                                  "Vertical.Acc_SD_18"
"Longitudinal.Acc_Max_100"                      "Longitudinal.Acc_Mean_23"
"Longitudinal.Acc_Min_107"                      "ankleX_Min_86"
"ankleX_Min_9"                                  "ankleX_Min_28"
"Lateral.Acc_RMS_125"                           "Vertical.Acc_SD_33"
"Skin.Temperature...IR.Thermometer...C._Mean_1" "Skin.Temperature...IR.Thermometer...C._RMS_91"
"ankleY_RMS_120"                                "ankleY_RMS_35"
"Lateral.Acc_Mean_128"                          "ankleZ_RMS_45"
"ankleX_RMS_40"                                 "Longitudinal.Acc_Mean_120"
"ankleX_Min_78"                                 "Skin.Temperature...IR.Thermometer...C._SD_6"
"Vertical.Acc_Max_23"                           "ankleZ_Min_79"
"Vertical.Acc_Min_44"                           "ECG.BR..rpm._Min_112"
"Vertical.Acc_SD_128"                           "Lateral.Acc_Mean_1"
"BR..rpm._Mean_1"                               "ankleZ_Min_23"
"Longitudinal.Acc_SD_11"                        "ankleX_RMS_51"
"ankleZ_Mean_28"                                "Vertical.Acc_Max_40"
"ankleY_Max_116"                                "Lateral.Acc_SD_112"
"Vertical.Acc_SD_39"                            "ankleY_Mean_1"
"ankleZ_Min_28"                                 "Lateral.Acc_SD_101"
"ankleZ_Min_101"                                "Vertical.Acc_SD_56"
"Longitudinal.Acc_SD_106"                       "ankleY_RMS_1"
"HR.Confidence_Mean_1"                          "Longitudinal.Acc_Mean_101"
"ankleX_Mean_1"                                 "ankleX_Mean_40"
"Vertical.Acc_SD_29"                            "ankleX_Max_97"
"BR.Confidence_Mean_1"                          "Lateral.Acc_Min_91"
"ECG.BR..rpm._Max_122"                          "HR..bpm._Mean_1"
"Longitudinal.Acc_SD_4"                         "ankleX_RMS_68"
```

Figure 5.2: The selected features from the ankle_equi combination built by Accordion.

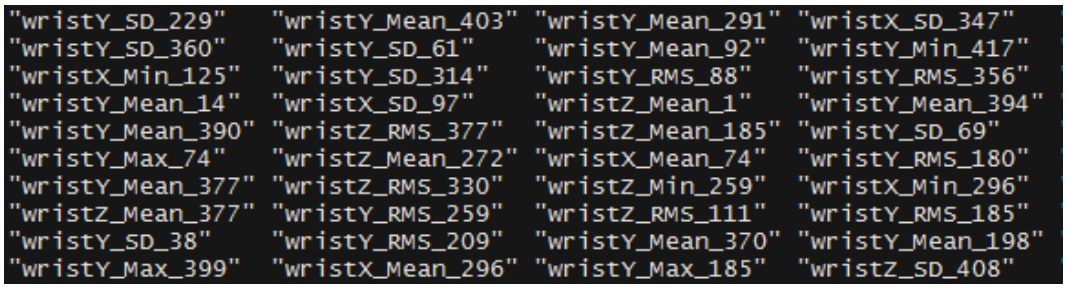

Figure 5.3: The selected features from the wrist combination built by Accordion.

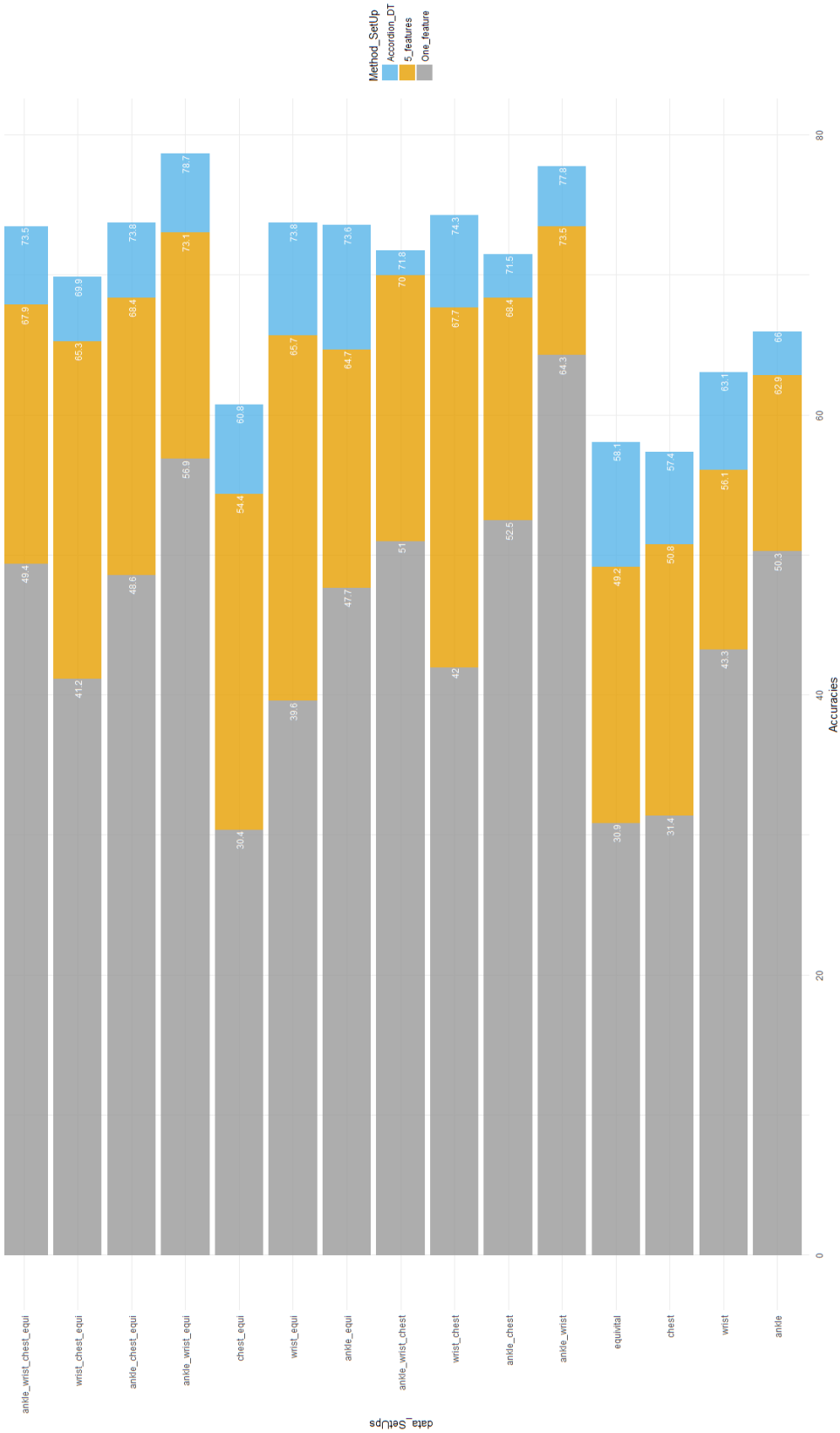

Figure 5.4: The models performance for every decision taken. Here, comparing the strategies of feature construction and selection for every sensor set-up.

5.3 Comparing Classifiers

Having decided, how which strategy will be used for feature construction and selection, our goal is to improve models' performance by using a better classifier. For that reason, we used an ensemble of trees, Random Forest, as it was introduced in theory.

In detail, looking at Table 5.4, it is clear that Random Forest method surpasses all the other models for every combination of sensors. Comparing the two models constructed by Accordion's features, it is visible, that they have better performance by more than 5% for every set-up, while for some of them exceeds the 10% (equi, chest_equi, wrist_chest_equi).

Table 5.4: The median accuracies of the LOPO evaluation for every model.

		Baseline (DT)		Accordion	
DataSetUps (28 participants)		*1 feature*	*5 features*	*DT*	*RF*
GA	ankle	50,3	62,9	66,0	71,0
	wrist	43,3	56,1	63,1	70,5
	chest	31,4	50,8	57,4	62,6
	ankle_wrist	64,3	73,5	77,8	83,0
	ankle_chest	52,5	68,4	71,5	77,5
	wrist_chest	42,0	67,7	74,3	80,6
	ankle_wrist_chest	51,0	70,0	71,8	78,5
EQ	equivital	30,9	49,2	58,1	69,5
GA&EQ	ankle_equi	47,7	64,7	73,6	79,8
	wrist_equi	39,6	65,7	73,8	81,9
	chest_equi	30,4	54,4	60,8	72,5
	ankle_wrist_equi	56,9	73,1	78,7	86,8
	ankle_chest_equi	48,6	68,4	73,8	81,8
	wrist_chest_equi	41,2	65,3	69,9	81,9
	ankle_wrist_chest_equi	49,4	67,9	73,5	82,6

In Figure 5.5, we can see in detail how the models improve their performance by using the random forest method on the Accordion features. Here, we can see that already, most models accuracy is around 75% with the chest_equi set-up having the greatest amount of improvement.

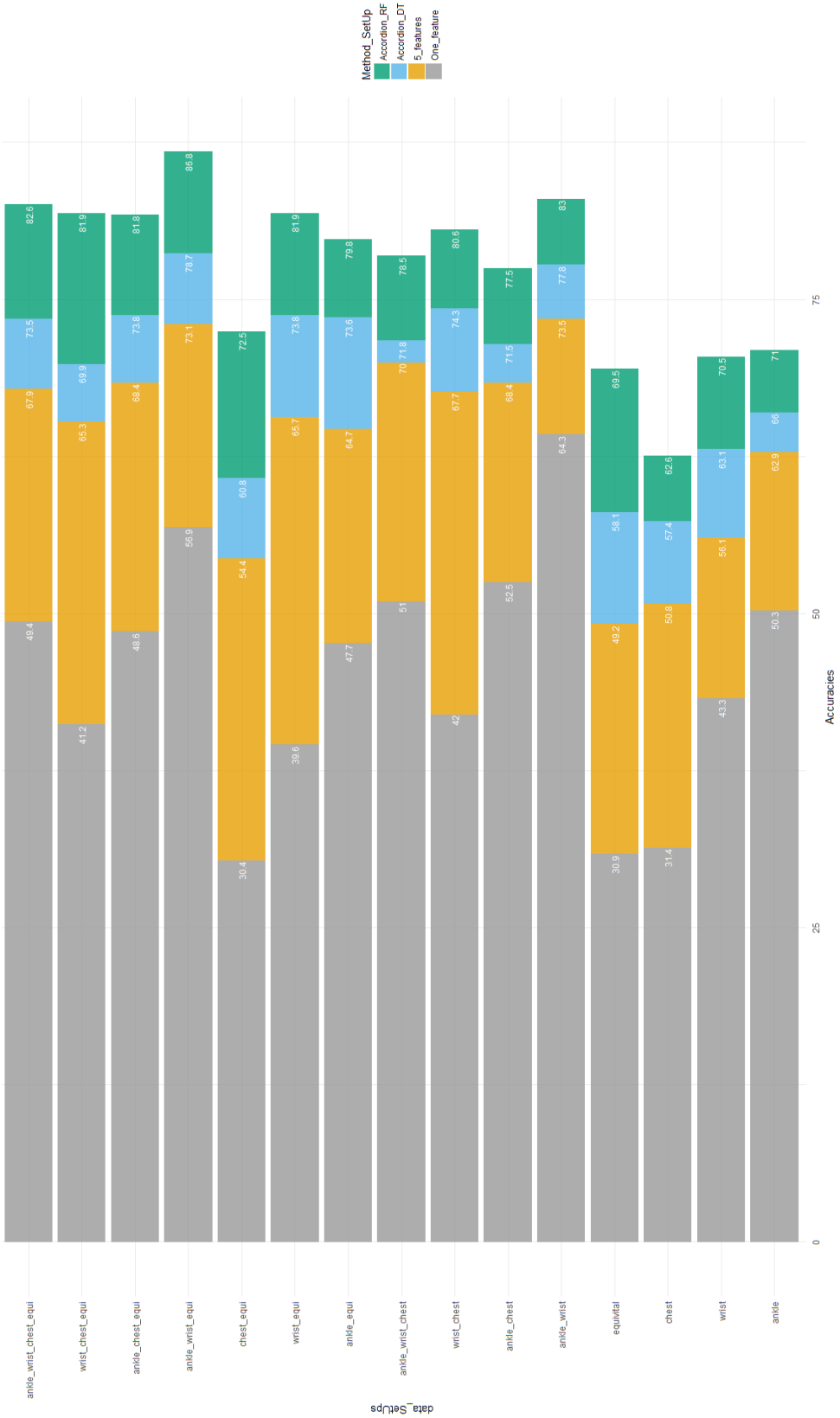

Figure 5.5: The models performance for every decision taken.

5.4 Sensor networks comparison

Analyzing the performance of different sensor set-ups in Table 5.4 and in Figure 5.5, the network which has the highest performance is the combination of ankle, wrist and equi sensors with an overall accuracy of 86.7% and the second one is the combination of ankle, wrist with 83%. This can also be seen in Figure 5.6, where the box-plots of the accuracy distributions for every set-up are displayed. It seems that combining accelerometers on ankle, wrist with physical measurements can lead to a pretty accurate activity recognition model. Nevertheless, combining accelerometer sensors on the wrists and ankles can give, already, satisfying results.

On the other hand, the model with the lower accuracy is the one of the accelerometer worn on the chest, with a median accuracy of 62.6%. This means, that it is harder to predict the activities by a stand alone chest accelerometer. The second worst model is the equivital alone, with a median accuracy of 69.6%. Here we have to note again that Equivital, except of the physical measurements, includes also an accelerometer. That means that having a belt on the chest combining accelerometers and physical measurements can improve the model already, something that is also more clear for the chest, equi set up (72.5%).

Another interesting point is that more data does not always mean a better model, since the sensor network with all the devices has lower performance than other combinations. Furthermore, it seems that in some cases adding chest accelerometers data does not improve or even decrease the activity recognition model's performance, e.g. ankle_wrist versus ankle_wrist_chest.

Finally, looking at the accuracy box-plots (Figure 5.6), we can observe that there are some outliers. Those lowest outliers, belong to the same participant's accuracy for all the sensors set-ups, participant GOTOV16, who seems to have a special walking pattern that our model could not predict. In all combinations that uses ankle and chest it produces the lowest accuracy, while for the model of wrist and wrist_equi the performance is above average, see Figures 5.7 & 5.8. The minimum accuracy can be found for the chest accelerometer with only 8.6%. Besides, the highest accuracy among all participants and sensor set-ups was performed by individual GOTOV14 with 99.14% for ankle_wrist combination. In Figure 5.9, the min and max performance for every participant is displayed.

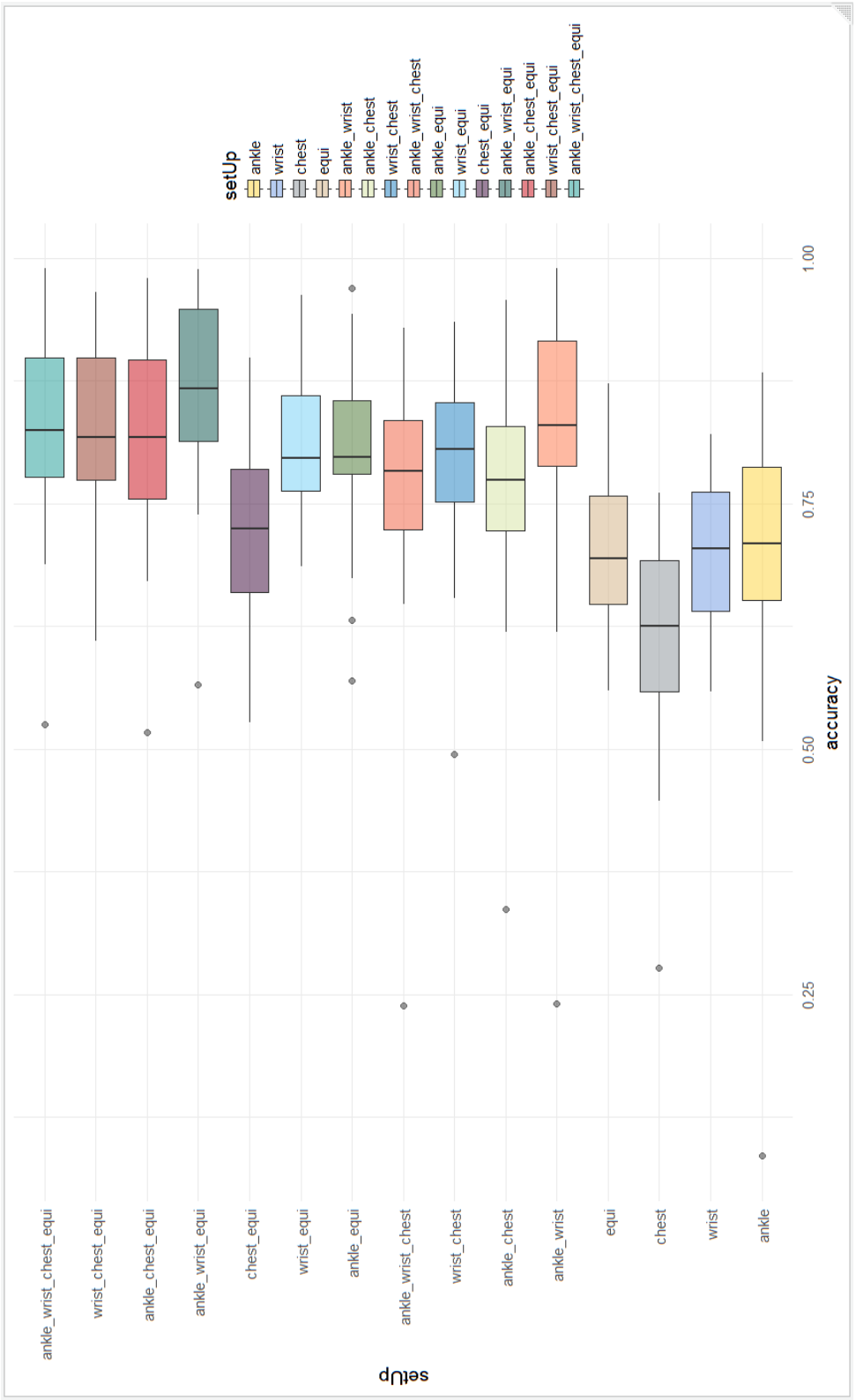

Figure 5.6: The box-plots of the accuracy distributions for every sensor network.

Participant	ankle	wrist	chest	ankle_wrist	ankle_chest	wrist_chest	ankle_wrist_chest
GOTOV05	78.3	76.6	59.3	85.1	63.8	75.7	75.5
GOTOV06	51.5	76.4	65.4	84.3	73.8	93.4	79.0
GOTOV07	81.1	58.9	69.2	79.6	75.2	83.1	76.2
GOTOV08	78.7	73.3	69.3	88.6	81.8	81.4	80.8
GOTOV10	71.2	76.9	44.8	91.8	78.4	80.2	74.4
GOTOV11	65.3	65.9	45.3	83.2	62.0	69.7	64.9
GOTOV13	78.8	75.4	66.3	93.0	83.4	84.1	83.8
GOTOV14	73.8	82.1	59.7	99.1	95.9	88.8	87.9
GOTOV15	57.1	58.0	67.7	77.8	73.2	76.5	70.1
GOTOV16	8.6	71.5	27.7	24.0	33.6	49.4	23.9
GOTOV17	70.8	73.3	60.2	79.5	76.0	76.1	76.0
GOTOV18	68.8	67.6	49.5	78.3	64.6	65.5	65.0
GOTOV20	69.5	56.0	71.5	79.6	78.6	89.4	78.0
GOTOV21	64.4	69.0	69.9	78.5	69.1	88.5	69.0
GOTOV22	83.4	80.6	60.4	91.5	83.8	85.2	81.8
GOTOV24	50.9	58.8	67.3	62.1	79.7	73.4	79.3
GOTOV25	85.4	79.9	68.4	95.7	86.8	93.6	86.3
GOTOV26	74.2	60.5	51.4	95.1	76.5	80.6	81.4
GOTOV27	70.1	61.4	76.2	73.7	81.4	86.5	84.9
GOTOV28	84.6	66.9	74.8	89.8	93.9	85.4	93.0
GOTOV29	64.4	69.5	62.3	83.0	82.6	78.2	82.3
GOTOV30	66.7	60.2	61.4	73.2	69.5	72.7	70.2
GOTOV31	64.9	67.5	76.2	79.3	82.8	80.4	84.0
GOTOV32	79.4	74.5	71.2	86.8	84.4	80.6	84.4
GOTOV33	74.6	64.9	56.2	81.5	71.0	73.6	68.5
GOTOV34	88.5	76.7	63.0	95.1	84.0	82.4	83.4
GOTOV35	75.4	74.1	55.0	79.0	72.6	69.1	73.0
GOTOV36	70.1	76.2	52.9	94.1	75.1	81.7	76.4

Figure 5.7: The predicted GENEActive sensor network accuracies per participant.

Participant	equi	ankle_equi	wrist_equi	chest_equi	ankle_wrist_equi	ankle_chest_equi	wrist_chest_equi	All_sensors
GOTOV05	64.6	82.2	82.1	64.5	89.1	73.6	89.5	83.4
GOTOV06	76.3	83.9	94.9	77.9	96.1	86.1	94.5	84.4
GOTOV07	63.0	79.1	78.9	73.0	82.8	83.6	79.3	82.2
GOTOV08	56.1	86.2	75.3	52.8	89.4	75.1	74.3	87.1
GOTOV10	67.8	78.6	85.1	70.3	96.3	77.3	83.1	77.4
GOTOV11	61.3	76.9	79.2	59.2	80.7	67.2	72.0	68.9
GOTOV13	82.5	86.2	96.4	85.7	97.6	96.8	96.0	96.3
GOTOV14	80.4	88.5	94.5	72.1	99.0	92.9	96.7	99.1
GOTOV15	63.6	79.7	93.0	78.6	87.4	82.0	93.3	80.6
GOTOV16	69.5	57.0	78.6	54.9	56.6	51.7	61.1	52.5
GOTOV17	62.4	74.7	68.7	62.1	76.6	75.6	66.9	77.7
GOTOV18	83.6	78.8	77.4	76.8	79.7	75.0	77.5	78.1
GOTOV20	71.1	82.3	72.4	80.5	84.6	91.1	91.3	89.8
GOTOV21	69.4	78.9	85.9	75.9	84.2	80.2	88.2	80.5
GOTOV22	87.4	94.4	95.7	90.0	98.5	96.8	86.2	96.6
GOTOV24	64.8	63.2	85.3	78.6	73.9	85.9	79.5	86.1
GOTOV25	75.6	96.9	76.6	88.6	97.1	98.0	79.4	97.7
GOTOV26	78.0	89.6	87.8	71.5	95.6	89.2	90.9	90.9
GOTOV27	73.9	77.3	82.4	64.5	79.6	81.6	82.0	78.1
GOTOV28	66.0	81.1	69.0	69.6	87.6	83.5	77.5	90.0
GOTOV29	71.6	85.2	78.9	72.9	86.3	82.1	78.2	74.6
GOTOV30	65.3	73.3	69.8	72.1	77.6	72.7	77.1	72.4
GOTOV31	75.4	79.7	73.2	72.2	85.2	81.8	77.3	81.8
GOTOV32	75.2	84.0	86.4	80.8	88.1	95.8	86.7	85.7
GOTOV33	69.2	78.2	75.5	66.2	81.6	76.6	73.8	77.7
GOTOV34	80.2	93.9	79.3	84.8	94.5	95.8	94.8	95.3
GOTOV35	69.6	80.0	84.0	73.2	86.2	80.9	81.7	83.0
GOTOV36	57.4	67.4	80.2	65.5	94.6	72.5	83.1	70.0

Figure 5.8: The predicted Equivital and GENEActive sensor network accuracies per participant.

Participant	Min accuracy setUp	Max accuracy setUp
GOTOV05	chest 59.3	wrist_chest_equi 89.5
GOTOV06	ankle 51.5	ankle_wrist_equi 96.15
GOTOV07	wrist 58.91	ankle_chest_equi 83.64
GOTOV08	chest_equi 52.82	ankle_wrist_equi 89.44
GOTOV10	chest 44.83	ankle_wrist_equi 96.27
GOTOV11	chest 45.28	ankle_wrist 83.15
GOTOV13	chest 66.27	ankle_wrist_equi 97.62
GOTOV14	chest 59.69	ankle_wrist 99.12
GOTOV15	ankle 57.1	wrist_chest_equi 93.29
GOTOV16	ankle 8.6	wrist_equi 78.59
GOTOV17	chest 60.22	ankle_wrist 79.54
GOTOV18	chest 49.54	equi 83.64
GOTOV20	wrist 55.98	wrist_chest_equi 91.33
GOTOV21	ankle 64.4	wrist_chest 88.49
GOTOV22	chest 60.45	ankle_wrist_equi 98.51
GOTOV24	ankle 50.9	All_sensors 86.14
GOTOV25	chest 68.41	ankle_chest_equi 98.02
GOTOV26	chest 51.35	ankle_wrist_equi 95.56
GOTOV27	wrist 61.43	wrist_chest 86.51
GOTOV28	equi 66.02	ankle_chest 93.87
GOTOV29	chest 62.31	ankle_wrist_equi 86.3
GOTOV30	wrist 60.18	ankle_wrist_equi 77.59
GOTOV31	ankle 64.9	ankle_wrist_equi 85.23
GOTOV32	chest 71.17	ankle_chest_equi 95.76
GOTOV33	chest 56.16	ankle_wrist_equi 81.64
GOTOV34	chest 63.01	ankle_chest_equi 95.79
GOTOV35	chest 55	ankle_wrist_equi 86.21
GOTOV36	chest 52.87	ankle_wrist_equi 94.56

Figure 5.9: The min and max predicted accuracies per participant.

Best Sensor network

In Figure 5.10, the accuracy distributions of the two best sensor networks are compared. It can be seen that both ankle_wrist and ankle_wrist_equi have similar distributions, however the combination with the physical measurements performs better in terms of overall accuracy.

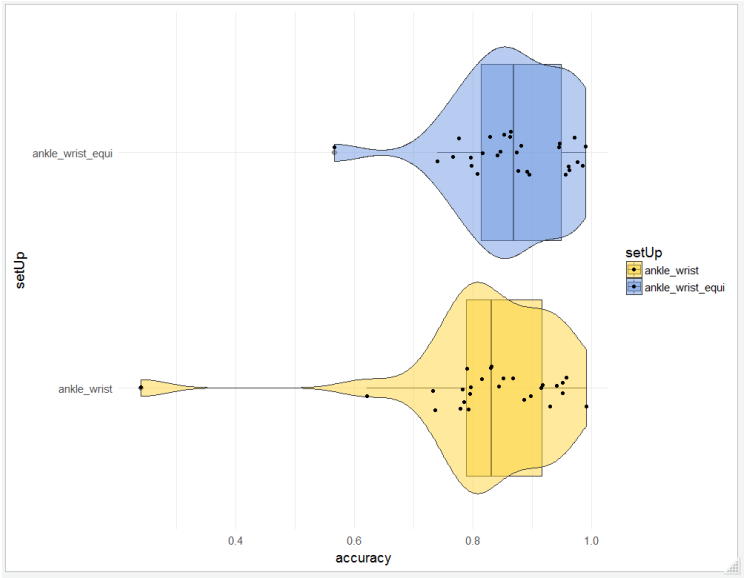

Figure 5.10: Box-plots of the two best sensor set-up.

Best Minimal Network

Concluding, we compare the minimal sensor networks. In Figure 5.11, a box-plot with the one sensor set-ups is presented. It can be seen, that ankle, wrist and equi outperform the chest, with a similar median accuracy. However, the distribution of wrist and equivital seem more concrete, since there are no outliers and most point are between the second and third quartile. Since, equivital combines both accelerometers and physical measurements can be chosen as the most suitable set-up using only one sensor.

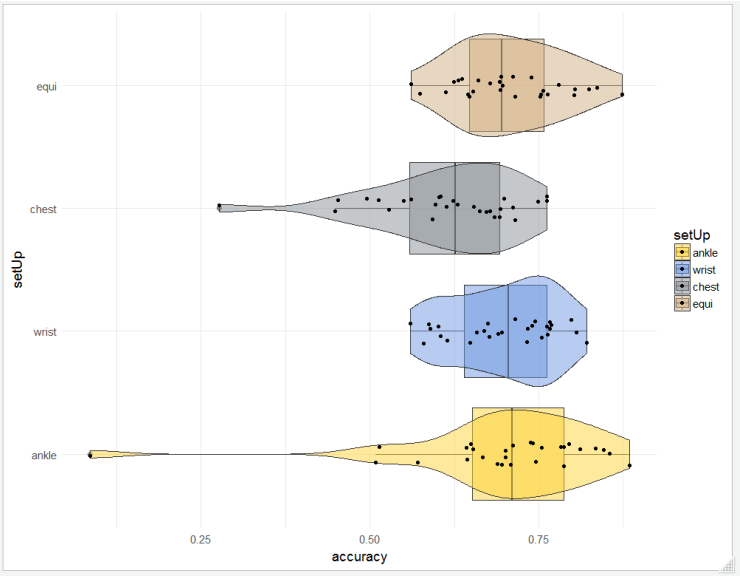

Figure 5.11: Box-plots comparing the best minimal set-up.

5.5 Activity analysis

In this section, we will, first, examine how different activities, per sensors set-up performed, for the Random Forest model. Afterwards, we will have a more detailed looked on them by examining the confusion matrices of some set-ups.

5.5.1 Comparing set-ups and activity predictions

In the following two Figures 5.12, 5.13, the accuracies of every activity per set-up are displayed. It can be clear that cycling, lying down and then jumping are the activities with the highest overall accuracies. Nevertheless, for cycling we have to consider that it is the class with the most training inputs. On the other hand, the activities that seem not performing really well are the different types of walking.

In more detail, for every set-up we can observe that the accuracies for the activities change, which is something we expect. Comparing, the ankle and the wrist combinations there are some interesting things, for example dishwashing is better

predicted from ankle (70% accuracy) than wrist (66% accuracy), while one would expect that since dishwashing is an activity using mainly the hands, it would be predicted easier from a wrist sensor. However, if we brake the activity in two parts, one for upper body and one for lowery, it can be that since dishwashing is a type of standing it is easier to be predicted from ankle, while the wrist might confuse it with other similar hand activities. Besides that, the wrist seems to distinguish standing a lot easier, 82% to 47%. This can be explained by the fact that the standing activity in our train set has no movement with hands at all (hands are probably next to the body) so there is no or too little signal input from the wrist sensor. On the other side, standing for the ankle, as we already stated, can be confused with other activities like dishwashing or sitting, since in both of those activities lower body is not moving. As a result, incorporating both sensors, those activities can be predicted with higher precision.

Furthermore, it can be observed that chest accelerometer, alone, is not able to distinguish the different activities easily. Therefore, there is also a decrease in terms of performance, for some models combined with chest, since the chest probably increases the confusion.

	ankle	wrist	chest	ankle & wrist	ankle & chest	wrist & chest	ankle & wrist & chest
Class	Precision	Precision	Precision	Precision	Precision	Precision	Precision
cycling	0.930	0.909	0.806	0.964	0.961	0.951	0.960
dishwashing	0.701	0.663	0.652	0.939	0.822	0.856	0.793
lyingDownLeft	0.907	0.852	0.880	0.936	0.957	0.891	0.957
lyingDownRight	0.765	0.839	0.877	0.819	0.933	0.883	0.923
sittingChair	0.577	0.725	0.407	0.872	0.531	0.807	0.519
sittingCouch	0.797	0.504	0.381	0.804	0.937	0.562	0.946
sittingSofa	0.562	0.587	0.442	0.912	0.542	0.642	0.507
stakingShelves	0.536	0.842	0.511	0.825	0.642	0.861	0.659
standing	0.470	0.823	0.245	0.850	0.649	0.861	0.633
step	0.761	0.548	0.392	0.814	0.739	0.567	0.728
syncJumping	0.525	0.817	0.673	0.777	0.909	0.877	0.937
vacuumCleaning	0.725	0.620	0.677	0.867	0.837	0.861	0.833
walkingFast	0.588	0.513	0.686	0.664	0.669	0.704	0.669
walkingNormal	0.398	0.402	0.534	0.417	0.397	0.553	0.443
walkingSlow	0.631	0.602	0.656	0.646	0.699	0.761	0.764
walkingStairsUp	0.349	0.355	0.105	0.296	0.338	0.562	0.360

Figure 5.12: The predicted activity accuracies for GENEActive sensor networks.

5.5.2 Confusion Matrices

Considering the above, it is essential to examine activities prediction dependency on set-ups, a bit deeper. To achieve that, we will explore the confusion matrices of ankle, wrist, equivital sensor networks and their combination ankle_wrist_equi.

Ankle

In Table 5.5, the ankle set-up is analyzed. Here we can see the things already discussed in the previous section. Standing is confused with different activities like *sitting* (Sofa, chair) and *dishwashing*. Moreover, it is hard, for an accelerometer worn in ankle, to differentiate between the different *sittings*, or households, *dishwashing*,

Class	equi	ankle & equi	wrist & equi	chest & equi	ankle & wrist & equi	ankle & chest & equi	wrist & chest & equi	all
	Precision	Precision	Precision	Precision	Precision	Precision	Precision	Precision
cycling	0.895	0.939	0.946	0.905	0.977	0.960	0.942	0.956
dishwashing	0.781	0.911	0.893	0.772	0.957	0.914	0.896	0.910
lyingDownLeft	0.927	0.938	0.969	0.926	0.937	0.961	0.923	0.955
lyingDownRight	0.949	0.968	0.945	0.962	0.922	0.931	0.901	0.931
sittingChair	0.450	0.663	0.925	0.560	0.852	0.737	0.903	0.675
sittingCouch	0.527	0.996	0.607	0.537	0.975	0.925	0.640	0.801
sittingSofa	0.547	0.689	0.703	0.485	0.910	0.749	0.738	0.632
stakingShelves	0.737	0.734	0.898	0.798	0.864	0.802	0.897	0.766
standing	0.433	0.548	0.838	0.510	0.809	0.584	0.826	0.551
step	0.460	0.776	0.581	0.485	0.864	0.867	0.568	0.806
syncJumping	0.792	0.950	0.907	0.530	0.951	0.948	0.908	0.943
vacuumCleaning	0.691	0.840	0.741	0.819	0.865	0.880	0.826	0.886
walkingFast	0.667	0.683	0.677	0.625	0.698	0.655	0.687	0.645
walkingNormal	0.487	0.520	0.549	0.470	0.514	0.452	0.558	0.452
walkingSlow	0.643	0.781	0.708	0.685	0.762	0.777	0.736	0.783
walkingStairsUp	0.320	0.411	0.583	0.253	0.307	0.338	0.727	0.363

Figure 5.13: The predicted activity accuracies for Equivital & GENEActive sensor networks.

stakingShelves, and *vacuumCleaning* are confused with each other. Finally, the higher confusion is taking place among the different types of *walking*. This is something anticipated, considering the fact that every individual has a different walking pace. Consequently, ones normal walking can be fast or slow walking for others.

Table 5.5: Confusion Matrix of ankle set-up, with red the max confusions per activity.

classes	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
cycling	1	14336	9	3	0	41	13	16	1	46	219	190	66	20	54	328	66
dishwashing	2	22	3818	2	0	28	0	164	781	587	7	0	17	1	8	6	2
lyingDownLeft	3	0	0	5782	185	2	343	1	32	27	0	0	0	0	0	0	1
lyingDownRight	4	37	186	186	5832	183	541	186	186	121	4	0	163	0	0	0	1
sittingChair	5	27	16	0	0	3263	16	2221	3	90	4	0	10	0	1	0	6
sittingCouch	6	747	0	313	222	5	5312	24	0	1	15	0	7	1	0	3	18
sittingSofa	7	13	80	0	0	1756	49	2780	91	127	1	0	30	5	0	7	10
stakingShelves	8	26	1616	1	0	39	6	130	3980	665	19	8	884	14	22	8	5
standing	9	47	507	0	0	733	8	630	217	2000	22	10	39	4	19	9	8
step	10	71	1	0	0	2	1	0	0	41	957	74	20	1	0	16	73
syncJumping	11	44	0	0	0	0	0	0	0	14	66	336	0	58	60	0	62
vacuumCleaning	12	147	77	4	0	22	27	55	1038	306	74	13	5101	11	32	102	30
walkingFast	13	11	0	0	0	0	0	0	0	0	0	0	0	4387	2483	573	1
walkingNormal	14	22	0	0	0	0	0	0	0	1	0	2	0	1861	2256	1524	1
walkingSlow	15	27	2	0	0	0	0	1	0	13	0	8	20	459	2275	4805	3
walkingStairsUp	16	13	10	0	0	0	1	0	1	7	51	60	0	232	197	15	315

Wrist

In Table 5.6, the wrist's confusion matrix is demonstrated. Contrasting this table with the one of the ankle, we can see that the confusions differ. First we observe that *dishwashing*'s higher confusion is with *cycling* and then a bit with other household activities, but it has a really low confusion with *standing*. Then we notice that the two sides *lying* have a slight confusion with each other, while the *sitting* have higher, with *sitting sofa* and *couch* being really connected. Lastly, different *walking* pace

have also the higher confusion, like in the ankle set up, with the difference that also the *step* class is largely mixed with *walkingSlow*.

Table 5.6: Confusion Matrix of wrist set-up, with red the max confusions per activity.

classes	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
cycling	1	**14290**	710	18	5	7	61	235	20	5	0	70	226	18	0	20	44
dishwashing	2	1108	**5013**	19	2	32	113	89	383	23	6	44	411	62	14	187	54
lyingDownLeft	3	12	13	**5071**	336	234	85	190	3	8	0	0	2	0	0	0	0
lyingDownRight	4	4	1	441	**4697**	73	334	49	1	0	0	0	0	0	0	0	0
sittingChair	5	61	15	482	335	**5011**	383	558	45	1	0	0	3	0	2	3	12
sittingCouch	6	106	61	148	785	559	**4125**	2211	148	19	24	0	1	0	0	4	0
sittingSofa	7	46	29	247	235	182	1177	**2909**	2	123	0	0	3	0	0	0	0
stakingShelves	8	86	145	10	29	115	154	44	**5758**	24	1	20	149	21	72	69	142
standing	9	11	14	10	0	0	33	95	18	**3574**	87	17	382	30	44	26	4
step	10	16	3	1	0	1	0	0	0	59	**893**	0	8	12	20	572	46
syncJumping	11	24	6	0	0	0	0	0	2	14	0	**428**	34	6	3	0	7
vacuumCleaning	12	601	451	27	0	24	37	13	97	199	23	95	**5180**	431	608	505	65
walkingFast	13	8	1	0	0	0	0	0	1	2	5	46	55	**3782**	2690	779	5
walkingNormal	14	6	1	0	0	0	1	0	4	6	19	6	62	2495	**2758**	1474	23
walkingSlow	15	20	19	2	0	4	3	1	12	47	375	3	51	507	1507	**4000**	97
walkingStairsUp	16	56	25	0	1	18	3	2	32	3	2	0	8	1	0	65	**119**

Equivital

In Table 5.7, the equivital's confusion matrix is given. Here we can notice that activities confusion is higher in terms of number of classes missclassified for every activity. Notwithstanding, there are similar high confusions with ankle and wrist, like *sitting* and *walking*. With different walking labels being again the ones with the larger mix.

Table 5.7: Confusion Matrix of equivital set-up, with red the max confusions per activity.

classes	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
cycling	1	**13455**	76	2	0	15	3	5	26	135	298	24	208	68	102	450	165
dishwashing	2	185	**4639**	0	0	174	36	28	372	197	12	0	261	13	12	4	6
lyingDownLeft	3	0	0	**5935**	462	0	0	1	0	3	0	0	0	0	0	0	0
lyingDownRight	4	0	0	293	**5545**	0	0	1	0	0	0	0	3	0	0	0	0
sittingChair	5	23	126	0	0	**2914**	1298	1496	100	490	5	0	3	0	10	5	7
sittingCouch	6	15	28	0	0	906	**2879**	1130	97	381	1	0	4	4	9	6	2
sittingSofa	7	0	25	0	3	1121	1094	**2902**	27	130	0	0	0	0	0	0	0
stakingShelves	8	129	548	0	0	179	129	32	**4797**	141	1	2	481	3	29	21	13
standing	9	69	479	4	0	632	814	550	306	**2361**	41	21	153	2	5	14	5
step	10	167	3	0	0	3	4	0	4	31	**800**	0	3	0	9	644	71
syncJumping	11	7	0	0	0	0	0	0	0	21	0	**588**	0	117	0	0	9
vacuumCleaning	12	1148	324	4	174	63	11	9	370	60	15	7	**5015**	6	3	32	19
walkingFast	13	38	1	0	0	0	0	0	0	3	0	46	0	**4242**	1894	100	39
walkingNormal	14	30	2	0	0	1	0	0	3	0	1	2	0	1975	**3450**	1588	29
walkingSlow	15	234	7	0	0	8	1	0	3	6	174	1	4	190	1493	**4058**	136
walkingStairsUp	16	19	7	0	0	4	1	0	0	1	35	5	4	16	22	105	**103**

Ankle-Wrist-Equivital

Finally yet importantly, in Table 5.8, the confusion matrix of the ankle-wrist-equivital network is presented. In this table, we can remark the lower confusion comparing to the other ones. We can notice the fact that *sitting* classes are now more clearly distinguished, something that is also happened for the household activities. Another interesting point is that *sittingCouch* and *cycling* seem to be in a way correlated, since the class of sitting couch highest confusion is with cycling. This confusion is probably introduced by the ankle sensor, as the confusion between them for ankle set-up is quite high. Yet, *walking* paces are highly mixed.

Table 5.8: Confusion Matrix of ankle, wrist and equivital set-up, with red the max confusions per activity.

classes	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
cycling	1	14244	11	1	0	12	47	36	2	24	10	27	43	4	7	52	65
dishwashing	2	19	5541	0	0	11	1	18	144	25	0	2	23	0	3	1	1
lyingDownLeft	3	0	0	5573	366	0	0	1	0	8	0	0	0	0	0	0	0
lyingDownRight	4	0	0	473	5636	0	0	4	0	0	0	0	0	0	0	0	0
sittingChair	5	24	33	0	0	5478	215	543	126	0	2	0	2	0	0	0	9
sittingCouch	6	111	0	0	0	4	5474	27	0	0	0	0	0	0	0	0	0
sittingSofa	7	33	12	0	0	126	272	4903	12	28	0	0	0	0	0	0	2
stakingShelves	8	6	212	0	0	191	36	181	5472	22	0	2	165	0	25	8	14
standing	9	20	17	3	0	2	6	250	18	3532	66	19	345	24	29	28	6
step	10	3	1	0	0	4	1	0	0	41	1199	1	1	5	6	62	64
syncJumping	11	4	0	0	0	0	0	0	0	19	0	604	0	1	0	0	7
vacuumCleaning	12	116	236	4	0	7	25	6	132	181	49	11	5349	4	14	46	6
walkingFast	13	11	0	0	0	0	0	0	0	0	0	2	0	4229	1655	166	0
walkingNormal	14	15	0	0	0	0	0	0	0	0	0	0	0	1602	3547	1739	0
walkingSlow	15	19	0	0	0	0	0	0	0	5	9	0	5	177	1156	4403	1
walkingStairsUp	16	33	19	0	0	5	3	1	0	2	54	12	2	282	292	213	407

5.6 Activity Ontology Trees

As we already noticed in Section 5.5.2, there are some activity classes that tend to be confused with each other. In order to have a higher understanding of them, we will build their hierarchy trees, based on the method introduced in Chapter 4.

Before we start building the trees, we will convert all confusion matrices to ones with the precision numbers per activity, like Table 5.9. This is done, as it was explained, in order to balance the inputs. To compute this table, we divide every activity row with the total number its inputs. Comparing now the Tables5.9 & 5.8 we can observe that while in Table 5.8 we meet the first max confusion in *walkingNormal*, for Table 5.9 we meet it in *walkingFast*.

In Figures 5.15, 5.16, 5.17, the ontology trees of wrist, equivital and their combination set up, ankle_wrist_equivital, are presented. All trees seem to have a similar structure, with two main subtrees created, the one with the walking classes and the one of sitting. The walking paces subtree, however, is the one constructed faster, since those classes have a high confusion.

Table 5.9: Confusion Matrix of ankle, wrist and equivital set-up normalized by the number of inputs for every activity, with red the max confusions per activity.

classes	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
cycling	1	**97,66**	0,08	0,01	0,00	0,08	0,32	0,25	0,01	0,17	0,07	0,19	0,30	0,03	0,05	0,36	0,45
dishwashing	2	0,33	**95,71**	0,00	0,00	0,19	0,02	0,31	2,49	0,43	0,00	0,04	0,40	0,00	0,05	0,02	0,02
lyingDownLeft	3	0,00	0,00	**93,69**	6,15	0,00	0,00	0,02	0,00	0,13	0,00	0,00	0,00	0,00	0,00	0,00	0,00
lyingDownRight	4	0,00	0,00	7,74	**92,19**	0,00	0,00	0,07	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
sittingChair	5	0,37	0,51	0,00	0,00	**85,16**	3,34	8,44	1,96	0,00	0,03	0,00	0,03	0,00	0,00	0,00	0,14
sittingCouch	6	1,98	0,00	0,00	0,00	0,07	**97,47**	0,48	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
sittingSofa	7	0,61	0,22	0,00	0,00	2,34	5,05	**90,99**	0,22	0,52	0,00	0,00	0,00	0,00	0,00	0,00	0,04
stakingShelves	8	0,10	3,35	0,00	0,00	3,02	0,57	2,86	**86,39**	0,35	0,00	0,03	2,61	0,00	0,40	0,13	0,22
standing	9	0,46	0,39	0,07	0,00	0,05	0,14	5,73	0,41	**80,91**	1,51	0,44	7,90	0,55	0,66	0,64	0,14
step	10	0,22	0,07	0,00	0,00	0,29	0,07	0,00	0,00	2,95	**86,38**	0,07	0,07	0,36	0,43	4,47	4,61
syncJumping	11	0,63	0,00	0,00	0,00	0,00	0,00	0,00	0,00	2,99	0,00	**95,11**	0,00	0,16	0,00	0,00	1,10
vacuumCleaning	12	1,88	3,82	0,07	0,00	0,11	0,40	0,10	2,13	2,93	0,79	0,18	**86,46**	0,07	0,23	0,74	0,10
walkingFast	13	0,18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03	0,00	**69,75**	27,30	2,74	0,00
walkingNormal	14	0,22	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	23,21	**51,38**	25,19	0,00
walkingSlow	15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,09	0,16	0,00	0,09	3,07	20,02	**76,24**	0,02
walkingStairsUp	16	2,49	1,43	0,00	0,00	0,38	0,23	0,08	0,00	0,15	4,08	0,91	0,15	21,28	22,03	16,07	**30,71**

Furthermore, for the ankle's ontology tree (Figure 5.14), it can be observed that the different walking paces are all combined with each other before the 10th iteration. An interesting point here is that *lyingdownLeft* and *lyingdownRight* are not merged in the same group. However, they are the last two labels to be merged with a group. As a result, *LyingDownRight* is merged in step 12 with the bigger group of *sitting*, *standing* and *households* which as a group has a lower confusion with *lyingDownLeft* than the group of *cycling* and *sittingCouch*. This is because if you have a look in ankles confusion matrix *lyingDownLeft* max confusion is with *sittingCouch*. On the other hand, for all the other combinations the two lying classes are combined with each other or in the same group (wrist).

The main point suggested, form the tree ontologies, is that the different walking paces should probably be merged to one class. In order to justify this point, the overall accuracy of the model per step is investigated. In Figure 5.18, it can be observed that in the first 5 steps where walking classes are merged for every set-up, the impact in the accuracy[1] is high. For example, for the ankle and wrist set-ups, from around 70% the model accuracy increases to $\approx 85\%$, almost 15% raise. Similarly, for the combinations of ankle-wrist and ankle-wrist-equi the increase is almost 10%. Particularly, for our best model (ankle-wrist-equi), combining the walking classes to one will increase the accuracy of our best model from 85% to 94%, which is a really satisfying accuracy for an activity recognition model predicting 12 labels.

Concluding, we compare the two ways of growing the tree ontology using the confusion matrix. In Figure 5.19, the non-normalized (greedy) and the normalized one, for ankle-wrist-equi, are compared. It is clear that merging in a greedy way

[1]The accuracy reported is the overall accuracy computed from the confusion matrix. For that reason, it may different $\pm 1\%$

increases the accuracy of the model faster. Nevertheless, also, in the greedy method the first 3 merges performed are the 3 different walking paces (slow, normal, fast) with a different order. This can be seen, as in the 3 merging both models have the same accuracy.

In Figure 5.20, the evolution of the accuracies when we merge the classes of walking paces to one for every set-up, can be seen.

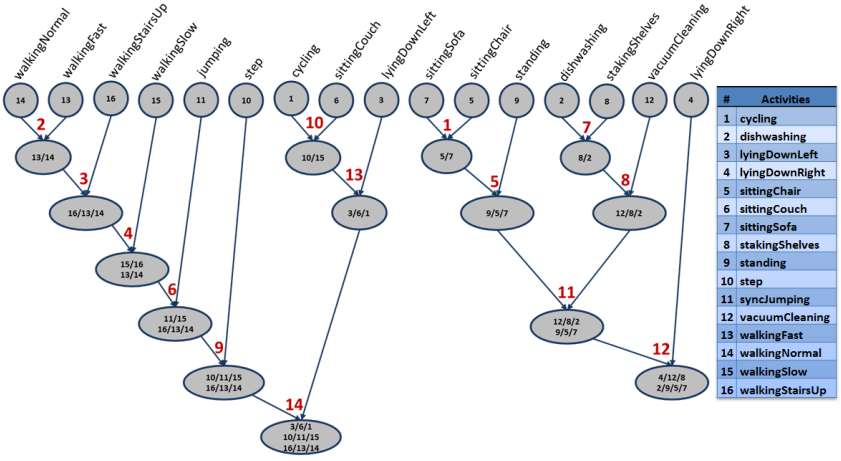

Figure 5.14: The ankle activity ontology tree, with red the order of merges.

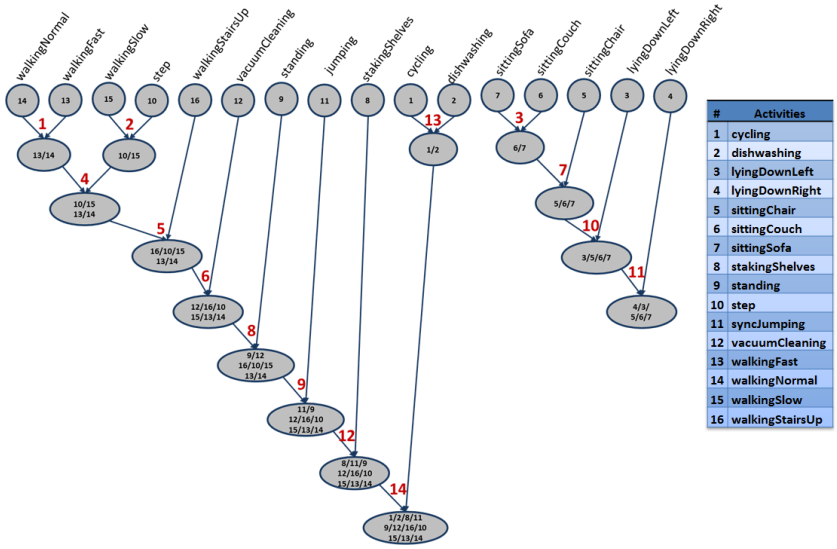

Figure 5.15: The wrist activity ontology tree, with red the order of merges.

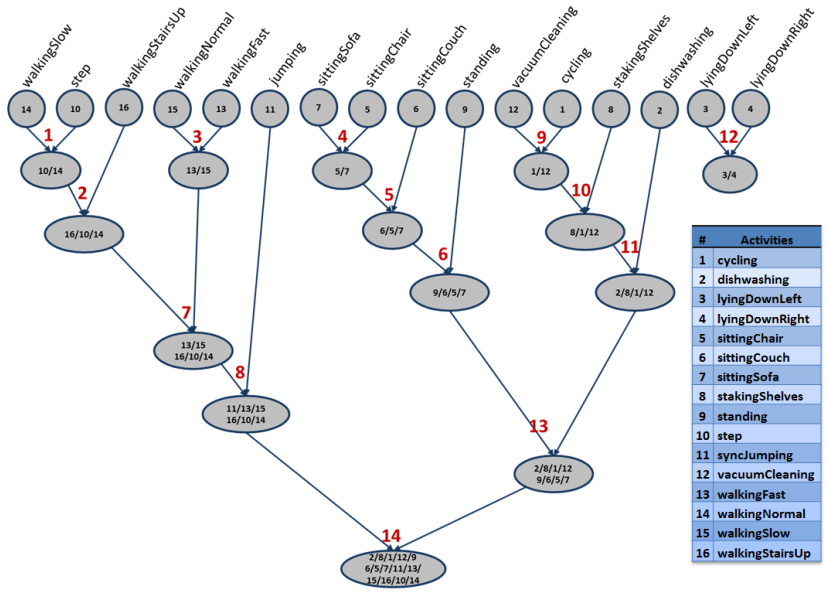

Figure 5.16: The equivital activity ontology tree, with red the order of merges.

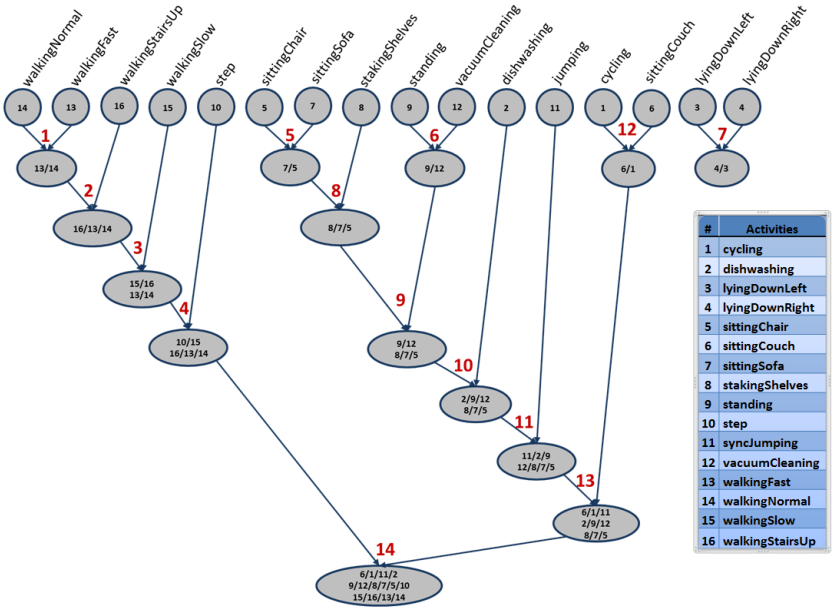

Figure 5.17: The ankle_wrist_equi activity ontology tree, with red the order of merges.

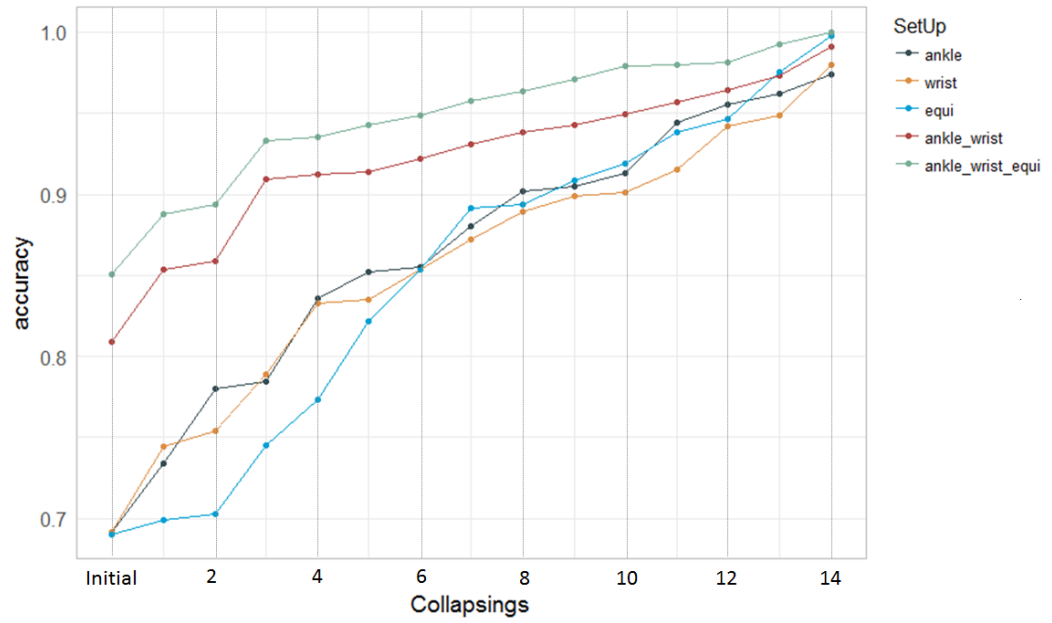

Figure 5.18: Model Accuracy in every step for ankle, wrist, equivital and their combination set-up.

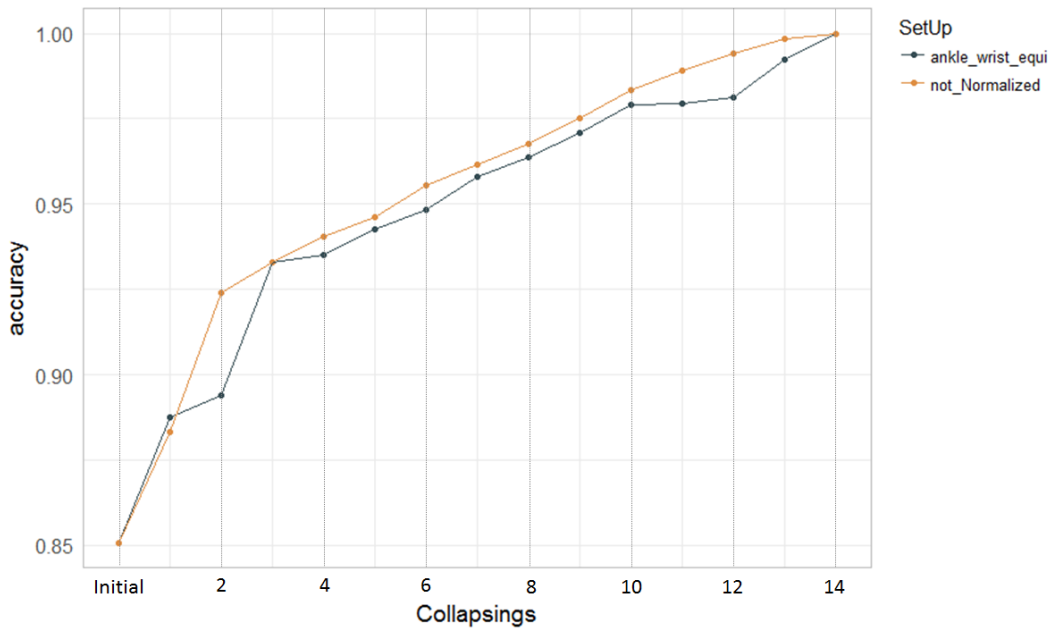

Figure 5.19: Comparing non normalized and normalized growing of activity ontology tree.

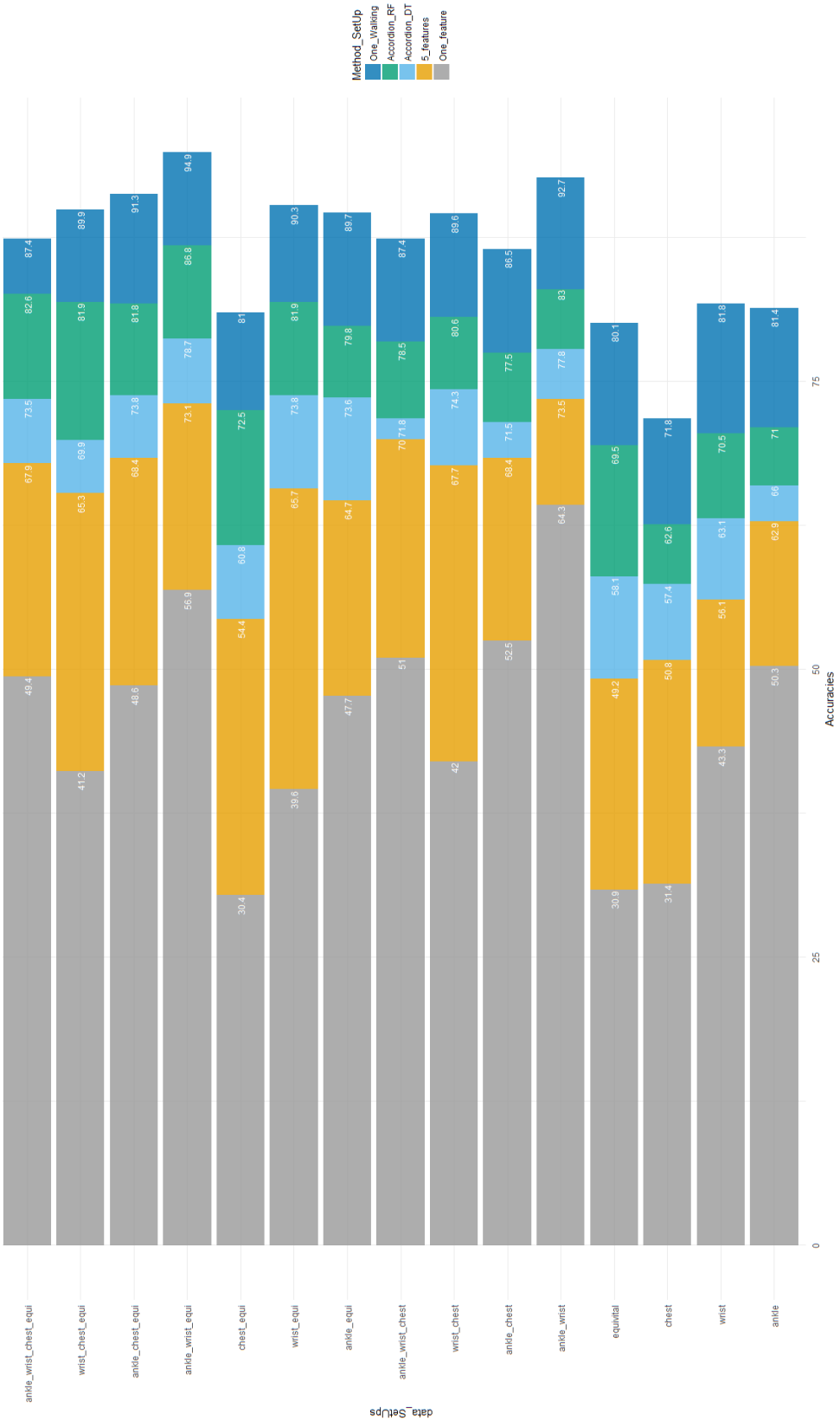

Figure 5.20: The models performance for every decision taken.

5.7 Temporal Nearest Neighbour Smoothing

In this section, we present the results of applying different temporal nearest neighbour smoothing filters on the Decision Tree build by Accordion for different sensor set-ups. Table 5.10, presents the experimental results. In the first column, the initial accuracy of every set-up is displayed and then the single and multiple smoothing filters are demonstrated.

For these experiments, we tested filters with window size from 3 to 31 seconds. On the single mode, we applied every filter and reported its accuracy, while for the multiple, we apply the one filter after the other starting from small ones to bigger.

As can be seen, both processes improved the overall accuracies of the model. Applying a single filter it seems that different windows between 11 and 27 seconds can produce the highest increases in prediction accuracy, of approximately 5%.

On the other hand, we achieve similar results by applying multiple smoothing filters, but only after the filters of 23 seconds and higher were applied one after the other.

The interesting point from these experiments is that single filters peak in earlier windows than the multiple, however without having a similar pattern for every set-up. On the other side, applying multiple filters one after the other it seems that after window of 19 seconds all set-ups start being around their peak and stabilizing there. Nevertheless, in order to prove this point more experiments should take place. Adding to that, we have to also point out that the main drawback of the multiple smoothing is that it is computationally more complex than a single filter pass.

Table 5.10: Table of Temporal Nearest Neighbour Smoothing results.

		Accordion LOPO	Temporal Nearest Neighborhood Smoothing															
			For every window								Adding in every window							
	DataSetUps		3	7	11	15	19	23	27	31	3	7	11	15	19	23	27	31
GA	ankle	66,0	68,6	70,4	71,4	72,4	69,9	67,2	65,9	65,3	68,6	69,8	71,2	71,1	71,3	71,2	71,9	71,9
	wrist	63,1	65,2	68,9	70,1	69,4	70,0	67,1	67,3	64,3	65,2	68,1	69,1	69,9	70,6	70,9	70,5	70,6
	chest	57,4	58,5	61,2	63,4	63,6	61,9	59,6	57,6	58,3	58,5	60,4	61,3	62,1	62,3	62,4	63,0	62,4
	ankle_wrist	77,8	78,4	80,2	81,1	81,4	82,0	81,9	82,2	80,8	78,4	79,7	80,8	81,2	81,6	81,6	81,8	81,6
	ankle_chest	71,5	72,9	74,5	74,8	74,7	75,2	75,8	74,7	72,9	72,9	74,2	75,3	75,9	75,8	76,5	75,8	75,9
	wrist_chest	74,3	75,6	79,1	79,3	79,4	79,4	78,0	75,1	74,5	75,6	77,5	79,2	79,1	79,9	80,1	80,2	79,7
	ankle_wrist_chest	70,6	71,6	73,2	74,7	75,0	73,9	71,9	69,5	68,8	71,6	73,2	74,0	74,0	73,6	73,6	74,7	74,8
EQ	equivital	58,1	59,6	61,7	62,0	62,7	62,5	60,5	59,3	59,5	59,6	61,6	62,5	63,5	64,1	64,4	64,6	64,8

Concluding, when this procedure was applied to the already refined models of random forest with 12 classes (one walking class), the increase in accuracy was $\approx 1\%$, with maximum 2% for the chest and equi_chest and with no significance (Kolmogorov-Smirnov's tests with $p>0.05$). This was expected since the activities confusion for every model have been minimized.

Chapter 6

Conclusion

The quest of higher accuracy

As it was discussed until now, the decisions of pre-processing, feature construction and selection, and method are crucial for the overall accuracy of the activity recognition model. All these choices and decisions have been exhaustively investigated through many papers in literature. In this thesis, we compared sensors and their body locations, different methods for feature construction and selection, two classifiers, and then we examined the influence of activity classes to the models performance.The goal for every of these steps was to improve the performance of the activity recognition model. In Figure 5.20, we presented exactly how every decision improved our model. Starting from a baselione model, with accuracy between 30% to 50%, and after different steps concluded to models between 81% to 95%, depending on the set-up.

6.1 Research questions

During this quest of higher accuracy and completing our objectives to analyze the role of sensor's placement we concluded to the answers of our main research questions, as they were presented in Section 1.5.

Best sensor network

Our main objective was to find the *best sensor network*. Throughout our experiments, it was clear that combining ankle and wrist models is the most efficient way to predict human activity. Physical measurements, can also give an extra boost to the accuracy but if there is no other need of them (e.g. energy expenditure prediction), it can be avoided it without significant difference.

Best minimal sensor network

Furthermore, we tried to conclude which *minimal sensor network* is the most efficient. Answering this question was not so clear, since except of chest sensor all the others; ankle, wrist, equivital, had an overall similar performance. Nevertheless, it was

proven that every one of them performs better for certain activities and so the choice of them depends on the set of activities that is our goal to predict.

Activity recognition model evaluation

Another really interesting point, was the evaluation of the models. Since recently, most researches were using n-folds cross validation for that. However, it is proven now that this evaluation creates overoptimistic results, as it is overfitting the models. We discussed and analyzed that this overfitting is result of the nature of the time series data and their autocorrelation. In order to have a more fair evaluation of the model, the most appropriate way would be the *Leave One Participant Out* or LOPO. This evaluation method tests the model on time-series data that were not used in the training data set. Moreover, it simulates the realistic scenario of predicting activities of an unknown individual. Overall, for physical activity monitoring - unless the development of personalized approaches is the explicit goal - subject independent validation techniques should be preferred [44].

Dealing with activity ontology

One of our last points, was the analyses of the activity classes and how they influence the prediction accuracy. Here, we try to create a data-driven activity ontology in order to understand which activities could be combined or not under a higher class. Throughout this investigation, we concluded, that having different forms of the same activity is not adding extra information, since it creates higher confusion. Therefore, activities such as different walking paces could be integrated under one higher class, e.g. walking.

Smoothing temporal predictions

Our final point, during this thesis, was to study how smoothing can further improve our models. As it was proven, smoothing can improve models, however the choices of window sizes, or applying single or multiple filters play a major role to the outcome. Adding to that, we noticed that the improvement to models that perform, already, really well is not significant enough.

6.2 GOTO study

Concluding, our motivation for this work was to create an activity recognition model for the GOTO study. In GOTO study, the sensors used were on the ankle and wrist. Therefore, the model with this network will be used to predict the activities. Furthermore, we will use the features constructed by Accordion and the random forest classifier. Finally, about activity classes, we will have only one class for walking.

Bibliography

[1] Bo Dong, Alexander Montoye, Rebecca Moore, Karin Pfeiffer, and Subir Biswas, *Energy-aware Activity Classification using Wearable Sensor Networks.* Proc SPIE, NIH Public Access, 2013.

[2] N. Noorit, N. Suvonvorn , and M. Karnchanadecha, *Model-based human action recognition.* Proc. Second International Conference on Digital Image Processing, International Society for Optics and Photonics, 2010.

[3] O. Lara and Labrador, *A survey on human activity recognition using wearable sensors.* Communications Surveys Tutorials, 2013.

[4] Yago Saez, Alejandro Baldominos and Pedro Isasi, *A Comparison Study of Classifier Algorithms for Cross-Person Physical Activity Recognition.* Sensors, 2017.

[5] Incel O, Kose M and Ersoy, *A review and taxonomy of activity recognition on mobile phones.* BioNanoScience, 2013.

[6] Yan, Z., Chakraborty, D., Mittal, S., Misra, A., and Aberer, K., *An exploration with online complex activity recognition using cellphone accelerometer.* UbiComp Adjunct, 2013.

[7] U. B. a. B. S. Andreas Bulling, *A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors.* ACM Computing Surveys (CSUR), 2014.

[8] Paillard-Borg S, Wang H-X, Winblad B, Fratiglioni L, *Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly.* IEEE Transactions on Biomedical Engineering, 2003.

[9] Paillard-Borg S, Wang H-X, Winblad B, Fratiglioni L, *Pattern of participationin leisure activities among older people in relation to their health conditionsand contextual factors: a survey in a Swedish urban area.* Ageing and Society, 2009.

[10] Van de Rest O, Schutte BAM, Deelen J, et al., *Metabolic effects of a 13-weeks lifestyle intervention in older adults: The Growing Old Together Study.* Aging (Albany NY), 2016.

[11] Curone D, Bertolotti G, Cristiani A, *A real-time and self-calibrating algorithmbased on triaxial accelerometer signals for the detection of human posture andactivity.* IEEE Transactions on Information Technology in Biomedicine, 2010.

[12] Parkka J, Ermes M, *Activity classification using realistic data from wearable sensors.* IEEE Transactions on Information Technology in Biomedicine, 2006.

[13] Vaughan IP, Ormerod SJ, *Improving the quality of distribution models for conservation by addressing shortcomings in the field collection of training data.* Conservation Biology, 2003.

[14] Lei Gao, A.K. Bourke, John Nelson, *Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems.* Medical Engineering and Physics, 2014.

[15] Natthapon Pannurat, Surapa Thiemjarus, Ekawit Nantajeewarawat, and Isara Anavantavrasilp, *Analysis of Optimal Sensor Positions for Activity Classification and Application on Different Data Scenario.* Sensors, 2017.

[16] A. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim, *A triaxial accelerometer based physical-activity recognition via augmented-signal features and a hierarchical recognizer.* IEEE Transactions on Information Technology in Biomedicine, 2010.

[17] Qi, X. Keally, M., Zhou, G., Li, Y., Ren, Z., *AdaSense: adapting sampling rates for activity recognition in body sensor networks.* In Proc. IEEE 19th Real-Time and Embedded Technology and Applications Symposium, 2013.

[18] Ferhat Attal, Sammer Mohammed, Mariam Debabrishvili, Faicel Chamroukhi, Latifa Oukhelou, and Yacine Amirat, *Physical Human Activity Recognition Using Wearable Sensors.* Sensors, 2015.

[19] Abdallah, Z.S., Gaber, M.M., Srinivasan, B., and Krishnaswamy, S., *CBARS: Cluster based classification for activity recognition systems.* In Proc. 1st Int. Conference on Advances Machine Learning Technologies, 2012.

[20] Brezmes, T., Gorricho, J.L., and Cotrina, J., *Activity recognition from accelerometer data on mobile phones.* In Proc. 10th International Work-Conference on Artificial Neural Networks, 2009.

[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer New York Inc., 2001.

[22] Hosam Rowaihy; Sharanya Eswaran; Matthew Johnson; Dinesh Verma; Amotz Bar-Noy; Theodore Brown; Thomas La Porta, *A survey of sensor selection schemes in wireless sensor networks.* SPIE Proceedings, 2007.

[23] Rami Debouk, Stephane Lafortune, Demosthenis Teneketzis, *On an Optimization Problem in Sensor Selection.* Discrete Event Dynamic Systems, 2002.

[24] Louis Atallah, Benny Lo, Rachel King and Guang-Zhong Yang, *Sensor Placement for Anctivity Detection using Wearable Accelerometers.* 2010 International Conference on Body Sensor Networks, 2011.

[25] Cleland, I.; Kikhia, B.; Nugent, C.; Boytsov, A.; Hallberg, J.; Synnes, K.; McClean, S.; Finlay, D., *Optimal Placement of Accelerometers for the Detection of Everyday Activities.* Sensors, 2013.

[26] Georgios Meditskos, Stamatia Dasiopoulou, Vasiliki Efstathiou, and Ioannis Kompatsiaris, *Ontology Patterns for Complex Activity Modelling.* Springer Berlin Heidelberg, 2013.

[27] Anindhita Dewabharata, Don Ming-Hui Wen, Shuo-Yan Chou, *An Activity Ontology for Context-Awared Health Promotion Application.* Computer Software and Applications Conference Workshops (COMPSACW), 2013.

[28] Amin Abdalla1, Yingjie Hu, David Carra, Naicong Li, Krzysztof Janowicz, *An ontology design pattern for activity reasoning.* Proceedings of the 5th Workshop on Ontology and Semantic Web Patterns (WOP2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, 2013.

[29] Quinlan RJ, *C4.5: Programs for Machine Learning.* Morgan Kaufmann Series in Machine Learning, 1993.

[30] Breiman, L, *Random Forests.* Machine Learning, 2001.

[31] M.F. Delgado, E. Cernadas, S. Barro, D. Amorim, *Do we Need Hundreds of Classifers to Solve Real World Classifcation Problems?* Journal of Machine Learning Research, 2014.

[32] L. Bao and S. S. Intille, *Activity recognition from user-annotated acceleration data.* Pervasive, 2004.

[33] Y. Hanai, J. Nishimura, and T. Kuroda, *Haar-like filtering for humanactivity recognition using 3d accelerometer.* IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009.

[34] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, *Activity classification using realistic data from wearable sensors.* IEEE Transactions on Information Technology in Biomedicine, 2006.

[35] Z.-Y. He and L.-W. Jin, *Activity recognition from acceleration data using ar model representation and svm.* International Conference on Machine Learning and Cybernetics, 2008.

[36] C. Zhu and W. Sheng, *Human daily activity recognition in robot assisted living using multi-sensor fusion.* IEEE International Conference on Robotics and Automation, 2009.

[37] Karantonis, D.M.; Narayanan, M.R.; Mathie, M.; Lovell, N.H.; Celler, B.G., *Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring.* IEEE Transactions on Information Technology in Biomedicine, 2009.

[38] Mathie, M.J., Celler, B.G., Lovell, N.H., Coster, A.C.F., *Classification of basic daily movements using a triaxial accelerometer.* Medical and Biological Engineering and Computing, 2004.

[39] Gjoreski, H.; Lustrek, M.; Gams, M., *Accelerometer Placement for Posture Recognition and Fall Detection.* In Proceedings of 2011 IEEE 7th International Conference on Intelligent Environments, Nottingham, UK, 2011.

[40] R. Cachucho, M. Meeng, U. Vespier, S. Nijssen, A. Knobbe, *Mining Multivariate Time Series with Mixed Sampling Rates.* ACM SIGKDD Explorations Newsletter, 2010.

[41] Zhengzheng Xing, Jian Pei, Eamonn Keogh, *A Brief Survey on Sequence Classification.* Proc. ACM UbiComp, 2014.

[42] M. W. Kadous, *Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series.* PhD thesis, 2002.

[43] Bedogni, L.; Di Felice, M.; Bononi, L., *By train or by car? Detecting the user's motion type through smartphone sensors data.* Proceedings of the 2012 IFIP Wireless Days (WD), 2012.

[44] A. Reiss, *Personalized Mobile Physical Activity Monitoring for Everyday Life.* PhD thesis, 2013.

[45] D. Minnen, T. Westeyn, T. Starner, J. Ward, and P. Lukowicz, *Performance metrics and evaluation issues for continuous activity recognition.* Performance Metrics for Intelligent Systems, 2006.

[46] Charles X. Ling, Jin Huang, and Harry Zhang, *AUC: A statistically consistent andmore discriminating measure than accuracy.* Proceedings of the 18th International Conference on Artificial Intelligence, 2003.

[47] Wikipedia, "Unix time." URL: `https://en.wikipedia.org/wiki/Unix_time#cite_note-4`, last checked on 2017-08-14.

[48] L. Rokach and O. Maimon, *Clustering methods.* Data mining and knowledge discovery handbook. Springer US, 2005.

[49] R Development Core Team, *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, 2008.

[50] B. C. Hornik K and Z. A, *Data mining and knowledge discovery handbook. Springer US.* Open-Source Machine Learning: R Meets Weka, Computational Statistics, 2009.

[51] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, 2002.

[52] M. K. C. from Jed Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, B. Kenkel, the R Core Team, M. Benesty, R. Lescarbeau, A. Ziem, L. Scrucca, Y. Tang, C. Candan, and T. Hunt., *caret: Classification and Regression Training*, 2017. R package version 6.0-76.