# Analysis of Activity Recognition and the Influence of Feature Extraction and Selection in an Android Based Device

Philip Holgersson, Fredrik Åkerberg

Master's thesis
2015:E50

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

CENTRUM SCIENTIARUM MATHEMATICARUM

# Analysis of Activity Recognition and the Influence of Feature Extraction and Selection in an Android Based Device

Philip Holgersson
philip.holgersson@gmail.com

Fredrik Åkerberg
j.f.akerberg@gmail.com

LUND UNIVERSITY

November 26, 2015

Centre for Mathematical Sciences
Faculty of Engineering

in collaboration with Sony Mobile Communication AB.

Supervisors:
Mårten Wadenbäck, LTH marten@maths.lth.se
Magnus Oskarsson, LTH magnuso@maths.lth.se
Christian Kullander, Sony Mobile christian.kullander@sonymobile.com

Examiner:

Kalle Åström, LTH kalle@maths.lth.se

**Abstract**

Tracking activities have lately been very popular in smartphones, which requires that the devices are able to classify activities correctly. Since this type of devices has limitations in both power consumption and computational performance, it is important to keep these factors to a minimum. Therefore, the low cost of the accelerometer sensor is a good platform to build a classification on. Although classifications that only use the accelerometer sensor is far from perfected, as far as accuracy is concerned.

To obtain a more accurate classification, it would be necessary to dissect the different parts of the classification process, and investigate if any of the parts can be improved. Many studies have been focusing on the different methods of calculating the classification, leading to many different well tested methods. However, very few have investigated the impact features may have on the classification, using the approach "more is better". Therefore this work focuses on feature selection combined with modified evaluation methods. Here we show that more features are not necessarily the best solution and that a modified naive evaluation method in most cases are better than a more recognized one. This can affect classifications in the future, especially since fewer features takes less power to compute. This is only the beginning, more studies are needed. We anticipate that our study will be used as a starting point for more in-depth studies in this field.



**Keywords**: accelerometer, feature extraction, features selection, least squares, classification, data analysis, confusion matrix, activity analysis

# Acknowledgements

We would like to send thanks to the the people that have helped us during the work with this master's thesis. First we would like to thank the people that have helped us collect the data needed to perform the analysis and those who have helped us with implementating the application. In addition we would like to send a special thanks to our supervisors at LTH, Mårten Wadenbäck and Magnus Oskarsson and to our supervisor at Sony Mobile Communications, Christian Kullander.

# Contents

# Chapter 1
# Introduction

As of today, our cell phones are more than just a communication device, they are small computers with capabilities to do advanced calculations which opens up for a lot of new areas of usage.

One of these areas is activity recognition, e.g. the device recognizes that you are running when you are out on your morning run. If the recognition is confident, it can be used in plenty of situations. If the data are analysed over a larger timeframe, it can be used to remind you that you have not moved in a couple of hours, and present the disadvantages if you continue not to move. Another case is if you run a lot, you can get information of new running shoes or other commercial things that are connected to your activity. It can also be used to track movement of elderly people, e.g. if a person has been completely still for a couple of hours, it can be a sign that this person has fallen, and relatives can be contacted [7]. If the data is analysed for a larger community, it can be used to increase the effectiveness of commuting, e.g. if the majority of the people in a city travel by bus between 7 and 8 o'clock, and close to none travel between 12 to 14, this can be used to change the time tables. But, as mentioned above, this requires that the recognition is confident and as of today, that is not always the case.

## The problem

To get a better activity recognition one can use different sensors in the phone. However, using many sensors in the phone at the same time for long periods will often drain the battery of the phone and might slow down the other functions of the device. Since battery and performance of cell phones is a hot topic in today's cell phone industry the use of continuous activity recognition aims to be as cheap as possible in both cost of battery power and computational performance. One approach is to use the expensive sensors, like GPS and gyro, as little as possible and use the much cheaper accelerometer sensor more. But how will this affect the performance in the activity recognition? This thesis is meant to be an early step in the effort of making the accelerometer responsible for more activity recognition and to see how much one can rely on its performance in accurate classification between different activities.
The questions this master thesis is trying to answer are therefore the following:

- Is it possible to get a good classification, when only using the accelerometer sensor?
- What effect on the classification does the placement of the phone have?
- Which set of features gives the best classification?
- Which are the problem areas between the three activities, Walking, Running and Bicycle?

# Chapter 2
# Background

This chapter contains the concepts on which the thesis has been based, and what the reader might need to know to understand the rest of the thesis.

## 2.1 Collecting data to analyse

In a study based on the analysis of data, the data collection part is an important one. Whatever method used for collecting the data, it is imperative to ensure that accurate data is collected to maintain the integrity of the research [36]. Inaccurate data collection can impact the results of a study negatively and lead to invalid or misleading results.

The data is often divided into training sets and evaluation sets, and for the training sets an important aspect is that it should make it possible for a classifier to discriminate between different classes in a correct way. How one collects the data can have a bigger impact in the classification than the method for classifying itself. For instance can a small set of data provide big problems in classifying and give a low accuracy [8, 4]. It is also important to keep a balance in the data between the different classes being analysed. It has been reported that an imbalanced data set can be difficult to use and classification becomes much easier when dealing with well-defined clusters of data [3]. The size of the training set has been shown to have a big impact on classification accuracy and researche's have also found a relationship between the size of the training set and the accuracy in classification [9].

For studies like the one in this master's thesis there are some environments one has to consider when collecting the data. Prior work such as Bao and Intille [2] writes about three such environments; a constrained laboratory setting with researchers monitoring the test subjects, an out-of-lab setting where the subject are responsible to collect and label the data themselves and a so called semi naturalistic setting where the subjects perform an activity that is disguised as another. An example that Bao and Intille uses is when they want the subject to collect data for "working on a computer" but tell them to use a computer to find the the worlds largest city and label the time it takes as goal for an imagnary obsticle coarse. In this way the subjects are less aware of the actual collection of data.

A constrained laboratory setting ensures that the data collected is correct and correctly labelled but a laboratory setting may also constrict or influence subject activity patterns in a way that will lower the accuracy when real world data for testing is run [2]. Bao and Intille [2] reports that data that has been collected for walking in a laboratory environment can show distinct phases of a consistent gait cycle which gives good training data, but data from the same subjects when not in the same environment might fluctuate in the gait phases and length of the gait pattern. This is one example on why it is important that the training data and the test data are collected in the same environment. Also, to simulate more realistic environments inside a lab is hard because one has to reconstruct the environments themselves to be able to do so [10].

Data that are collected and labelled only by the subjects themselves, in an uncontrolled out-of-lab setting, as in e.g. the research of Rai et al. [29], might yield an increased amount of data. However, this type of collection has a higher probability for being recalled due to the fact that subject might not remember what activity they performed or will provide wrong labelled data which will decrease the accuracy of the classification with general test data [10].

A semi-controlled out-of-lab setting where the subjects are collecting data in a naturalistic environment but are told exactly how to perform the activities, or if the subject are the researchers themselves, one might experience the same problems as with a constrained laboratory setting. Even if the algorithm is very accurate it might give false values in a live version although the training data were collected out-of-lab. The data might show the same distinct patterns as it would when collected in a laboratory setting and would suffer when there are more variations in the patterns like when the activity is performed more naturally [2].

The problems above can be compared to the development of new technologies. When developing and testing a product in a laboratory setting it has a high rate of failure when released into the natural environment. How humans behave in natural settings has a strong connection to the setting itself, and products will have a higher probability to fail when they are not developed according to how people behave in the real world as supposed to how one expects them to behave from testing in a laboratory setting [10].

## 2.2 Features

Features are important concepts in data analysis. The use of features is crucial in determining differences in the data. The data is divided into pieces that contain more concentrated information of it that will hopefully give it more distinct characteristics, while unnecessary information is discarded. A feature extracted from the data is such a piece of information and can be used when classifying activities as in this thesis. To create a feature from the data, one applys different methods and mathematical calculations to the data and new numbers are formed. These can hopefully be used to separate one activity from another.

From an accelerometer used in this master's thesis three initial values can be obtained, the $X - axis$, the $Y - axis$ and the $Z - axis$. Depending on what type of activity is being performed these inputs have different values, values that are used when putting them into different classes i.e. classifying them. However, these three might not provide sufficient information as they are to separate the activities and that is when the use of features comes into play. Features are built, or calculated, from the initial three values to get more differences between the activities being performed as described above. To make algorithms in pattern recognition and classification effective and useable, it is imperative to choose features that are informative, independent and discriminant [38].

### 2.2.1 Feature extraction

Data from an accelerometer sensor can be difficult to use directly and may need some pre-processing stages to meet the time frame on some physical activities [7]. Extracting features from the data is such a process. To address the problem there are a large number of techniques that one can use in extracting features. The different techniques can be divided in different domains such as time domain, frequency domain and what can be called discrete representation domains. Some techniques, or methods, of extracting features rely on the ability to transform input signals to and from the different domains [7]. For each domain there are numerous algorithms and methods to generate features for analysis. The methods based on the frequency domain have been used in large extent to extract repetitive parts of a signal such as the periodic nature of certain activities. To transform a signal from the time domain to the frequency domain the Fast Fourier Transform (FFT) is often used [7].

### 2.2.2 Feature selection

After extracting a number of features it is a good idea to select those that have a larger impact on the classification than others and to reduce the feature space optimally [1]. Furthermore, it might be beneficial for some applications to use a subset of features instead of trying to use all of them. Benefits of selecting a subset of the features include saving the cost of calculating features that do not affect the result or that adds unwanted noise [14]. The set of features extracted might also be too large to manage [38] or just not adding to the accuracy of the classification at all due to the correlation that exists in the collected data and, consequently, between extracted features [1]. If the features are going to be computed on a mobile device, one has to take into account the limitations in computational performance and battery power that exist. To be able to reduce the computational cost, reduce the time of the classification process, and to increase the accuracy of the classification, feature selection is a very important part in data analysis [1]. Feature selection is also used for simplifying models, shortening training times and enhanced generalizations [39].

There are numerous different ways in which feature selection can be done. There are many algorithms and filters designed to select the best subset of features. Principal component analysis (PCA), is one method that has been used before, [32, 14] Correlation-based Feature Selection (CFS) and ReliefF are other methods used in similar projects [1]. However, it is possible to use a more naive approach to select the best subset of features as Ravi et al. [30] gives a small example of. These have got both advantages and disadvantages compared to non naive methods depending on the complexity of, and knowledge about, the non-naive methods, such as that they might be easier to set up and use but be more time consuming when performing.

## 2.3 Related work

There are a great deal studies done in the field of accelerometer data analysis and it is difficult to relate to all the material available. The previous works that we have studied for this master's thesis covers most of the areas in the field and many of them do so quite extensively. They range from analysis performed in the late 1990's up to more recent ones which we think give good grounds in covering the most in the field. Some of the previous works have analysed the area in different ways or angles and based on the material used in this thesis we believe that we bring a new aspect to the studies of accelerometer data analysis.

This thesis differs from the previous works in a number of aspects. Some, like Bao and Intille [2], have used biaxial accelerometers while others have used the tri-axial accelerometer sensor [30] as we do. Most of the studies were performed using multiple accelerometers that are only used for research to collect their data whereas only the accelerometer sensor found in an ordinary Android based cell phone is used in this work. In that aspect, Kwapisz, Weiss, and Moores [11] research resembles this master's thesis in several ways. They used an Android device, i.e. one tri-axial sensor, with a self-made application to collect their data. Both Lekland and Sabanovic [13] and Weiss and Lockhart [34] collected their data in this manner. Kwapisz, Weiss, and Moore did however only use one placement of the device when collecting their data while we, e.g. Bao and Intille [2], examine five different placements. In those works where an Android device have been used, in particular a cell phone running Android, none have addressed the issue of the orientation of the device, which is something this master's thesis does.

Since the classification presented in this paper is meant to be used in an cell phone, the computational cost and performance have to be considered, which is something Figo et al. [7] have done by examining the performance that is needed for the calculation of various features.

Most of the previous works perform their classifications and algorithms on a large amount of activities whereas only three activities are used in this one. Except from Lekland and Sabanovic who concentrated their work solely on activities in the gym, others use more everyday activities like standing, sitting, walking, running, climbing stairs etc. while this thesis only covers the activities walking, running and biking.

For our work we use a modification of the least squares method found in Bishop [5] for the classification of activities, a method few or any of the previous work have used. Most others have used either the kNN, Naïve Bayes or SVM method, some uses all these three and some use even more methods of classification [1, 2, 13, 29, 30].

The most significant difference this master's thesis have from the previous work is however a deeper examination of the features used when performing an activity classification. As mentioned, the concept of features is an important part of analysis of this type and all of the prior work have used features, feature extraction and feature selection in one way or another. Although the amount of used features varies, most of the prior work use features in different domain areas [1, 7, 29, 30, 31]. However, none of them have calculated features across their natural domains in the same way that this master's thesis has done and very few have used features from other analysis areas. Optimizing the feature space with selecting the best subset of features is another aspect that differs this paper from most of the prior work. Examinations of the area have been done but while Lu et al. [14], who in the image data analysis field presented a method for reducing the feature space by choosing the subset of features with the most information, being one of them that have done the most, few others have examined the phenomenon as extensively as this thesis does.

## 2.4 Division of work

Most of the work has been done together or in close collaboration with each other but Fredrik has been more involved in setting up the analysis environment in matlab and with the implementation and work with the classifier while Philip has been more involved in the work with the data collecting application and in extracting features from the collected data.

## 2.5 Statistical classification

In statistics, classification is the problem of determining to which category (class) a new observation belongs. The classification is based on a training set where the class is known. For this study it is the problem of determining to which class (Walking, Running or Bicycle) a specific amount of accelerometer data belongs. The algorithm that solves the classification is known as a classifier.

To get a more accurate classification, it is often needed to modify the original values to make every class more distinct. A possible way to do this is to evaluate what qualities in the data can make a clear difference between different classes, these values are called features [40].

## 2.6 Classification with Least Squares

| **Definitions for the section** | |
|---|---|
| ***X*** | The feature matrix where the last column is 1 |
| ***k*** | number of classes |
| ***r*** | number of rows in $X$ |
| ***n*** | length of $X_i$ |
| ***t*** | either 1 or 0, but only one 1 on each row. |

For each class being classified, one can use the following function:

$$y_i = a_i^T x + d_i = x^T a_i + d_i = \begin{bmatrix} x^T & 1 \end{bmatrix} \begin{bmatrix} a_i \\ d_i \end{bmatrix} \tag{2.1}$$

This can be rewritten as:

$$T = Y = X\widetilde{W} \tag{2.2}$$

where $T$ is based on the training data, exactly how this is done will be described later. $\widetilde{W}$ is the solution from least squares of the equation (2.2), which is used to evalutate new data.

### 2.6.1 Training classification on one class

For training dat $Y$ is never really used, since the method is solved with the known $T$ instead. For the training data, the *T-vector* contains 1 and 0. For each row in the *X-matrix*, the corresponding row in the *T-vector* gets the value 1 if the row is the right class, and 0 if it is another class. The appearance of this vector is as follows:

$$T = \begin{pmatrix} t_1 \\ \vdots \\ t_r \end{pmatrix} \tag{2.3}$$

Every row in $X$-matrix contains a vector of features ending with a 1. Basically this is the data that the classification are based on.

$$X = \begin{pmatrix} f_1^T & 1 \\ \vdots & \vdots \\ f_k^T & 1 \end{pmatrix} \tag{2.4}$$

where $f = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^T$ is the feature vector.

$\widetilde{W}$ is the result of this method, and the values can be used to see which features are more significant for the classification. The larger values of $||[a_1^1, \ldots, a_1^n]||$ are more significant, and the smaller are less significant.

$$\widetilde{W} = \begin{pmatrix} a_{11} \\ \vdots \\ a_{1n} \\ d_1 \end{pmatrix} \tag{2.5}$$

## 2.6.2 Training classification on N classes

The difference for the $T$ when you have many classes, is that you have a matrix instead of an array. For each row, instead of only determining if the row in the *X-matrix* belongs to the class or not, each row in the *T-matrix* has a 1 for the index of the correct class and 0 for every other index. E.g. if we have three classes: $\begin{bmatrix} class_1 & class_2 & class_3 \end{bmatrix}$ if $X_i$ belongs to $class_2$, $T_i = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$

$$T = \begin{pmatrix} t_{11} & \ldots & t_{k1} \\ \vdots & \ddots & \vdots \\ t_{1r} & \ldots & t_{kr} \end{pmatrix} \tag{2.6}$$

For $\widetilde{W}$, the difference is that it is a matrix instead of an array, and each row corresponds to a class.

$$\widetilde{W} = \begin{pmatrix} a_{11} & \ldots & a_{k1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \ldots & a_{kn} \\ d_1 & \ldots & d_n \end{pmatrix} \tag{2.7}$$

## 2.6.3 Classifying

To evaluate the classification, the following equation is used.

$$\widetilde{S} = \widetilde{W}^T \begin{bmatrix} \widehat{X}^T & 1 \end{bmatrix} \tag{2.8}$$

where $\widehat{X}$ is the feature vector that will be classified, and $\widetilde{S}$ is the classification. Depending on how many classes, the appearance of $\widetilde{S}$ will different. The appearance is as follows, in the case with $n$ classes:

$$\widehat{C} = \begin{bmatrix} class_1 & \ldots & class_k \end{bmatrix} \rightarrow \widetilde{S} = \begin{bmatrix} s_1 & \ldots & s_k \end{bmatrix} \tag{2.9}$$

In $\widetilde{S}$ there will exist at least one value, where $s_i > 0$. The index of this $s_i$ corresponds to the index of the class($class_i$) in $\widehat{C}$ which the classifier has decided to be the correct one, if there are more $s_i > 0$, we decided to mark this as inconclusive.

## 2.7 The Confusion Matrix

The Confusion Matrix is a method to visualize how many, true positives, false positives, true negatives and false negatives with e.g. classification, and conclude if the classifier confuses a class with another class. In classification this gives the possibility to see exactly which classes that are hard to classify correctly. An example of the appearance of a confusion matrix is as follows:

| | **c1** | **c2** | **c3** |
|---|---|---|---|
| **c1** | 95% | 0% | 5% |
| **c2** | 10% | 80% | 0% |
| **c3** | 15% | 15% | 70% |

Here one can see that for class c3, 70% are true positives, and the other 30% as false negatives, where 15% are classified as class c2, and 15% are classified as class c1. [35]

# Chapter 3
# Methodology

In this chapter the methodology of the work is presented. It is divided in the different aspects of the work that have been covered. It describes what has been done during the work of this project, how it has been done and why we chose to do it in that way.

## 3.1 Collecting data

We chose to collect the data in a natural environment with data collected both by ourselves and from subjects not involved in the actual project. We chose to collect the data in this way to ensure that the data were as natural as possible. If we would have collected the data only by ourselves, the result might have been misleading. This is because even if we tried to perform the activities as normally as possible there are still going to be some times when we subconsciously performed the activity to try to produce “good” data and this could not be classified as an completely naturalistic environment as mentioned earlier in section 2.1 according to the study that Bao and Intille [2] have done. Further referring to Bao and Intille, subjects not involved in the project would not have the same thoughts and would perform the activity more natural as we did not specifically tell them to collect the data while performing an activity in a certain way, we only specified that they should collect the data at minimum of five minutes at a time to ensure that enough amount of motion repetitions were performed and to ensure that the files could generate sufficient amount of scopes. Had the data only been collected for e.g. 30 seconds, most of the data would be ignored, due to the implementation of the feature extraction and classification, and the result had been invalid. The subjects were also told that they should place the device on one of the different placements we wanted to examine. The placements can be seen in table 3.1 on the next page.

We also used multiple subjects to help collect the data to get the range of the data better in order to try to keep the data as balanced as possible [10].

The purpose of this project is to examine accelerometer data from a general perspective

**Table 3.1:** The five placements used in this study

- Front pocket
- Back pocket
- Upper arm
- Bag (preferably worn on the back)
- Hand

and not for certain individuals. Therefore we felt that the result would be more reliable with many different subjects collecting the data. The goal was to collect data from around 20 subjects, which we almost reached. Data where subjects were running became somewhat of a hindrance and we had some problems in collecting data from that activity. Since we did not want to collect the main part of the data ourselves due to facts mentioned earlier in section 2.1, we had to rely on volunteers to help us collect data and most of them either walked or rode a bike for the most part.

### 3.1.1 Android application

The application for collecting data used was developed in collaboration with Sony Mobile Communication AB in Lund. It is written for the Android platform and is supported by the latest version of the platform.

When a subject has finished performing an activity, the application creates a text file *(.txt)* containing a header, information of what activity Google's activity recognition engine thought was performed and raw data from the accelerometer sensor. The header contained a user-ID of the person using the application, which activity and placement the user told the application that they were performing and the duration of the activity. Each text file that was created was named with the user-ID and the timestamp in milliseconds of when the activity was started. Figure 3.1 illustrates how the accelerometer data was stored in the *.txt*-files created by the application.

```
1440675756816 -12.272079 -7.1431007 4.953005
1440675756835 -7.9152308 -9.642648 -0.6434417
1440675756856 -7.5680714 -11.717822 1.3006501
1440675756877 -9.339781 -11.018715 -1.7244238
1440675756897 -5.856816 -8.80767 1.1869255
1440675756915 -2.2750902 -9.770737 1.3521254
1440675756935 -1.8321629 -6.670246 -1.0468647
1440675756957 -2.3732524 -5.468955 1.0851719
1440675756976 -1.7423803 -3.0202854 -1.5466543
1440675756996 -1.8483237 -1.6573862 -0.0077811554
1440675757017 -1.8860323 -1.7280151 0.23882163
1440675757036 -2.7814639 -3.5895069 2.794632
1440675757056 -1.0462662 -5.728726 6.8198833
1440675757077 -0.46088383 -7.1191587 6.941988
1440675757097 -1.18094 -5.9005103 0.7930793
1440675757116 -1.3820529 -6.579266 1.282095
1440675757137 -1.22224 -16.196177 3.0298622
1440675757157 2.4384944 -19.6133 1.3856442
1440675757176 -1.614889 -14.036008 -0.5793968
```

**Figure 3.1:** Picture showing how the accelerometer data looks like when collected. The four columns are timestamp, X-axis, Y-axis and Z-axis

When a user collected data with the application they first chose the activity and then specified their user-ID and the placement of the device. The id was stored for future use, meaning it only had to be entered once. Figure 3.2 illustrates how the user-ID was entered and the placement selected.

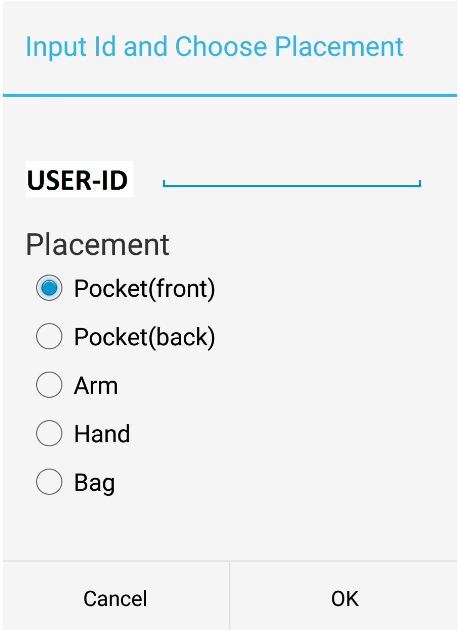


**Figure 3.2:** Picture showing the dialog for entering user-ID and selecting placement

## 3.2 Finding problematic areas

One step in this project was to find which areas, which was most problematic with this type of classification. We started to look at what Google's engine gave as result when performing the different activities with different placements of the device. For this we used a confusion matrix as described in section 2.7. Our confusion matrix had a slightly different appearance from the original appearance. The appearance was as follows:

| | **Walk** | **Run** | **Bike** | **Ignored** | **NaN** |
|---|---|---|---|---|---|
| **Walk** | 90% | 5% | 5% | 0% | 0% |
| **Run** | 10% | 85% | 5% | 0% | 0% |
| **Bike** | 3% | 2% | 95% | 0% | 0% |

*Walk*: Percentage of files, where the engine says the activity is walking.

*Run*: Percentage of files, where the engine says the activity is running.

*Bike*: Percentage of files, where the engine says the activity is bicycling.

*Ignored*: Percentage of files, where the engine says the activity is "ignore".

*NaN*: Percentage of files, where the engine data was missing.

Since the engine usually did not have only one activity, we decided to calculate the percentage of every activity the file contained. To decide which activity the whole file represented, we used the activity that had the greatest number of instances. After our own classifications was done, we created confusion matrices of our results. We used the different matrices to compare, primarily the problem areas of our classifications towards the engine, to see if our were better in any way or if we had the same problems. We also used the percentages to evaluate if our classifications were better or worse than the those of the enginge.

## 3.3 Setting up the analysis environment

We used Matlab for most of the analysis, because of its advantages considering manipulation of matrices. This section describes how the different parts of the analysis when bound together. Every part is described broadly, but will be described more in depth later in the report. Initially we analysed the files from the data collection, and separated header, engine-inputs, and accelerometer data. The accelerometer data for each file was then used to calculate a number of scopes. For each scope a feature vector was calculated, and for each file this resulted in a feature matrix, where every row was a feature vector. When all files were analysed, we concatenated all feature matrices for each placement and corresponding activity, and also one feature matrix for each activity containing all placements for the specific activity.

As mentioned by Intille et al.[10], and previously in section 2.1, a balanced training set is more prone to give better classifications. A balanced training set is a training set with equal parts of each class. Therefore we tested both balanced and unbalanced sets.

When all the feature matrices were calculated, it was time to train the classifier. We decided to use half of each feature matrix as training data, and the rest as evaluation data. The training data was used to calculate the classification with help from the least squares method described in section 2.6. We calculated four different types of classification methods, the first one, was a three-class classification, where all the classes was classified against each other. The other three classifications were two-class classification, one for every pair between the three activities.

After the classifications were calculated, we evaluated the classifications with the evaluation data. To get an understandable result, we did this for every feature vector in the evaluation data, and summarized the result in a new vector. This vector was afterwards divided by the number of feature vectors, to get the percentages and the appearance as follows:

$$\begin{bmatrix} \%Walking & \%Running & \%Bicycle & \%Inconsistent \end{bmatrix} \tag{3.1}$$

We used three different methods to evaluate the classifications. The first one used the three-class classification, whereas the second and third, were variants of the two-class classification, these methods will be described later.

After the initial evaluation, we evaluated which features were best. We did this by iterating through the feature matrix and did the above for each unique pair of features. We then saved the results in a matrix, containing the result vectors from each of the methods and the index of the features in the feature vector. This matrix was used to select the best pairs, which we combined to create a vector with the best features, sorted from best to worst. We did this method for each of the evaluation methods (Multiclass, Cross Guess, and Cross). We noticed that the results from Cross Guess and Cross were identical, which led to that we only did this for Multiclass and Cross Guess. We used another method to evaluate which features was best, we used the values in $\widetilde{W}$, as mentioned in section 2.6.2, to determine which features were more significant for the classification. This also resulted in a vector with the best features for this method. The result from this methods was three vectors that contained the number of all features, sorted from best to worst, for each of the three methods.

We redid the whole process with the two best features, followed by the three best and so on. For each of these we calculated the mean of the percentage of the true positives for each of the three classification methods. This resulted in 6 different versions described in the table below:

| | Method | Balanced | Classification |
|---|---|---|---|
| PWM B | Pairwise Iteration | Yes | Multiclass |
| PWC B | Pairwise Iteration | Yes | Cross Guess |
| WA B | $\widetilde{W}$ | Yes | Multiclass |
| PWM | Pairwise Iteration | No | Multiclass |
| PWC | Pairwise Iteration | No | Cross Guess |
| WA | $\widetilde{W}$ | No | Multiclass |

We plotted the result of the mean percentage of the classification for 2 to 170 of the best features, for each of the version in the table above. These plots were used to see how many features were needed to get a good classification for each of the versions. After this, we reran the program again for each of the above versions and with the new feature matrices, containing the features we decided were best for each of the versions. From the result we determined which versions that gave the best classification.

## 3.4 Features

As mentioned before in 2.3 this project differs from previous work much for the focus on features we have had. Aside from performing an analysis of the accelerometer data in the aspect of activity recognition, we wanted to see how a more elaborate approach to features would affect the result. The main goal with the features was to find the set of features that would give the best end result of the classification. We also examined the amount of features that would give a better result. Would it have any effect, and if so how much, if we used 5, 10 or 100 features to classify and which combination of features in the different sizes would give the best result?

To further explore the features we not only focused on features normally used in this type of work, but features that are used when analysing other type of data. The fields of sound- and image analysis were the two main fields that were explored. There were some promising candidates found during this research, but unfortunately most of them were too hard to implement in a way that they could be used to full extent, or even at all. Total Harmonic Distortion (THD) and Zero Crossing Rate (ZCR) were two of the ones that we were able to utilize, much because they had very simple predefined functions implemented in Matlab with the possibility to use an array as the only input and a single value as output. They are both mostly used in data analysis involving wave functions such as sound [7] or heart sound [31] and they were both interesting to use and to get results from, as described in sections 3.4.4.8 and 3.4.4.9.

### 3.4.1 Scopes

When loading a file with accelerometer data the first and last 1000 sample points were omitted. This was done since there might be mislabelling due to fluctuations in the data coming from starting the application and putting the device in the placement and to stop the recording at the end of the activity session [30]. These fluctuations could cause the classification to fail and give misleading results. The rest of the data was divided into a number of scopes. Each scope was represented by approximately 10 seconds of data with a 50% overlap (5 seconds) among the scopes. The overlap was chosen since it has been proven to give good results before [1, 2, 30]. The size of the scopes was chosen to cover a sufficient amount of data so that the movement cycles in the activity could be repeated a sufficient number of times and long enough that there would not be too few scopes, also a scope time of 10 seconds has been proven to be a good choice before [11].

During the creation of the scopes some of the data files made the program crash due to faults in the accelerometer sensor which stopped providing data and thus creating gaps of several seconds in the timestamps. To solve the problem the decision was made to skip the entire scope. After creating the scopes, the features were extracted to produce a so called feature vector for each scope.

### 3.4.2 Orientation invariance

During the work of extracting features we performed tests to see how the features worked together. We then noticed that in some cases the X-axis and Y-axis would have the same type of characteristics and it was easy to find differences in two different classes while the Z-axis would give opposite values. For example if the X-axis and Y-axis would have large values for running and small vlaues for walking, the Z-axis had a large value for walking and a small for running. In other cases it was the Y-axis that was different from the other two. Since we had the ambition to find and use only the best features in the end this was a problem. This meant that we could not decide whether a test file belonged to one class or another based only on the three axes. When discussing the problem we concluded that accelerometer values can give different values depending on the orientation of the device. Performing the same activity with the same placement may give completely different values depending on the orientation of the phone. If the device is turned 90 degrees, the same activity and placement can give the same values for the X-axis as the Y-axis for another orientation. To try to solve the problem we started to think of a way to combine the three axes into one to make the result the same every time. We created five different combinations of the three inputs and calculated features based solely on these.

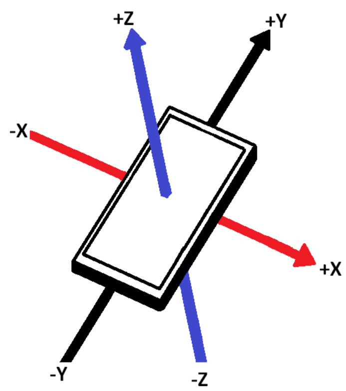


**Figure 3.3:** Figure illustrating the original orientation of the $X$-, $Y$- and $Z$-axes on a mobile device

During examination of these features we realised that it was only one of the five that was valid to have and that was the magnitude i.e.

$$\sqrt{x^2 + y^2 + z^2}. \tag{3.2}$$

Discussing the matter further the idea of removing the importance of the orientation of the phone arose, i.e making the orientation invariant. In this way it would not matter whether the phone was orientated in one way or another, the values would still be the same.

In order to make the orientation invariant the three inputs, X, Y and Z were taken as a column vector $A = \begin{pmatrix} x_i & y_i & z_i \end{pmatrix}^T$. From this a new input vector, called $\widetilde{A}$, has to be found such as $\widetilde{A} = RA$, where $R$ denotes a $3-by-3$ rotation matrix with $R^T R = I$ and $det(R) = I$. Even if the rotation matrix is unknown the values in $A$ will not be affected in size, only in orientation, since $\widetilde{A}^T \widetilde{A} = A^T R^T \cdot RA = A^T A$.

Finding $\widetilde{A}$ was done in three steps. In the first step the column vector $A$ is multiplyied by its own transpose, a row vector, to create a $3-by-3$-matrix $M$ such as $A^T A = M$, which is independent of $R$. The matrix $M$ is then made into a $6-by-1$ column vector containing the unique inputs from $M$. The index $i$ was denoted as an input sample i.e. each row in the accelerometer data file.

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \begin{pmatrix} x_i & y_i & z_i \end{pmatrix} = \begin{pmatrix} x_i^2 & x_i y_i & x_i z_i \\ x_i y_i & y_i^2 & y_i z_i \\ x_i z_i & y_i z_i & z_i^2 \end{pmatrix} \rightarrow \begin{pmatrix} x_i^2 \\ x_i y_i \\ x_i z_i \\ y_i^2 \\ y_i z_i \\ z_i^2 \end{pmatrix} = U_i \tag{3.3}$$

This is done for every row of accelerometer data in the file. The resulting $6-by-1$ column vector in this step is called $U$. The next step in the process were to normalize every row by taking the sum of the $\alpha$ number of columns before and after and then divide by $2 \cdot \alpha + 1$, where $\alpha = 3$ in this case. This step produce a new vector called the *U-hat-vector* ($\hat{U}$) from the *U-vector*.

$$\hat{U}_j = \frac{\sum_{i=j-\alpha}^{j+\alpha} U_i}{2 \cdot \alpha + 1} \tag{3.4}$$

The last step in the process reverts half the first step, e.g. go from a 6-by-1 vector to a 3-by-3 matrix. This matrix was called the *S-matrix*.

$$S_i = \begin{pmatrix} \hat{U}_i(1) & \hat{U}_i(2) & \hat{U}_i(3) \\ \hat{U}_i(2) & \hat{U}_i(4) & \hat{U}_i(5) \\ \hat{U}_i(3) & \hat{U}_i(5) & \hat{U}_i(6) \end{pmatrix} \tag{3.5}$$

The eigenvalues and eigenvectors of $S_i$ are then calculated from this *S-matrix* such as $S_i V_i = E_i V_i$ using the matlab command $[E_i, V_i] = \text{eig}(S_i)$, were $E_i$ is a matrix with the eigenvalues in the diagonal and $V_i$ is a matrix with the eigenvectors in the columns. From the eigenvalue matrix, new values for $x_i$, $y_i$ and $z_i$, henceforth called $\hat{x}_i$, $\hat{y}_i$- and $\hat{z}_i$, were derived by taking $\hat{x}_i = E_i(1,1)$, $\hat{y}_i = E_i(2,2)$ and $\hat{z}_i = E_i(3,3)$.

### 3.4.3 Normalizing features

There is no guarantee that all the inputs are good. Some inputs may falter and the sensor will send abnormal values. In order to reduce the effect that this had on the rest of the features, each feature was normalized. This minimized the weight on the features, hence minimizing the influence that one abnormal value would have on the rest of the same features. The normalization was done by, for every value in a column in the feature matrix, subtracting the mean value of the column, $mean(f)$, from the column and divide this by the standard deviation of the column, $std(f)$, as in equation (3.6).

$$f_i = \frac{f_i - mean(f)}{std(f)} \tag{3.6}$$

where $f$ is a column in the feature matrix and $i$ has the range from 1 to the length of that column. Each column in the feature matrix corresponds to one of the features in the feature vector.

### 3.4.4 Feature extraction

Feature extraction has been a big part in this project. As mentioned in section 3.4 we did not only look at features generally used for accelerometer data, but also other fields.

In this master's thesis work features were extracted from both the time- and frequency domain as mentioned in section 2.2.1. Apart from extracting features from the $\hat{X}$, $\hat{Y}$, $\hat{Z}$ and the combination of the three[1] a lot of the features were converted to the frequency domain by applying the Fast Fourier Transform (*FFT*) [20] and some were converted using the Discrete Cosine Transform (*DCT*) [19]. The *FFT* and the *DCT* were applied to all the inputs mentioned above and to further explore the possibilities with the *FFT* it was used in four different ways. These were the original value, the real part [25], the absolute value, $|a + bi| = \sqrt{a^2 + b^2}$, [15] and the original value multiplied with its own conjugate value, $Z \cdot \overline{Z} = |Z|^2$, [17] of the *FFT*.

For computational efficiency, the *FFT* requires the number of data points in a window to be a power of 2 [1]. Hence, both the *FFT* and the *DCT* were calculated on the form $y = function^2(x, n)$ where $n$ was calculated as $n = 2^{nextpow2(L)}$ with $L = length(\hat{X})$ i.e. the length of the input signal. This was done to pad $y$ with trailing zeroes to have a lenght of power of 2 [20].

A total of 16 values were generated from the *FFT* and 4 from the *DCT*. Together with 4 initial values, $\hat{X}$, $\hat{Y}$, $\hat{Z}$ and the magnitude *(MAG)*, there were 24 values that were used to calculate the features from.

[1]Magnitude referring to equation (3.2)
[2]function referring to $\mathrm{FFT}$- and $\mathrm{DCT}$-functions in Matlab

**Table 3.2:** Table showing the values that the features were calculated from

| | | | |
|---|---|---|---|
| 1 | $\hat{X}$ | 13 | $\mathrm{abs}(\mathrm{FFT}(\hat{X}))$ |
| 2 | $\hat{Y}$ | 14 | $\mathrm{abs}(\mathrm{FFT}(\hat{Y}))$ |
| 3 | $\hat{Z}$ | 15 | $\mathrm{abs}(\mathrm{FFT}(\hat{Z}))$ |
| 4 | *MAG* | 16 | $\mathrm{abs}(\mathrm{FFT}(\mathit{MAG}))$ |
| 5 | $\mathrm{FFT}(\hat{X})$ | 17 | $\mathrm{FFT}(\hat{X})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{X}))$ |
| 6 | $\mathrm{FFT}(\hat{Y})$ | 18 | $\mathrm{FFT}(\hat{Y})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{Y}))$ |
| 7 | $\mathrm{FFT}(\hat{Z})$ | 19 | $\mathrm{FFT}(\hat{Z})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{Z}))$ |
| 8 | $\mathrm{FFT}(\mathit{MAG})$ | 20 | $\mathrm{FFT}(\mathit{MAG})\cdot\mathrm{conj}(\mathrm{FFT}(\mathit{MAG}))$ |
| 9 | $\mathrm{real}(\mathrm{FFT}(\hat{X}))$ | 21 | $\mathrm{DCT}(\hat{X})$ |
| 10 | $\mathrm{real}(\mathrm{FFT}(\hat{Y}))$ | 22 | $\mathrm{DCT}(\hat{Y})$ |
| 11 | $\mathrm{real}(\mathrm{FFT}(\hat{Z}))$ | 33 | $\mathrm{DCT}(\hat{Z})$ |
| 12 | $\mathrm{real}(\mathrm{FFT}(\mathit{MAG}))$ | 24 | $\mathrm{DCT}(\mathit{MAG})$ |

From these 24 the following feature categories were calculated:

- Mean [16, 1]
- Variance [28, 7, 1]
- Eigenvalues of the covariance [18]
- Max value [23]
- Energy [7, 1]
- Entropy [7]
- Mean of the three most recurrent values(peak) [1]
- Zero Crossing Rate [43, 42, 7, 31]
- Total Harmonic Distortion [27, 41, 31]

Some features are often only used in one domain [30], but since we have chosen to investigate features further, the decision were made to calculate the most of the feature categories across the domain boundaries. The energy and entropy are examples of features that are often only calculated in the frequency domain but in this paper they were calculated using all of the 24 values in table 3.2.

#### 3.4.4.1 Mean

The mean value of the input was chosen as a feature as it has been proven to give accurate results in some postures and activities [2] and is used in this type of analysis very often. Using the mean in this situation is meaningful since it is calculated with small computational cost and with minimal memory requirements. Removing spikes and other noises are other reasons why it is used [7]. However, for this master's thesis work, it were only used on the first eight values shown in table 3.2, the $\hat{X}$, $\hat{Y}$, $\hat{Z}$ and the magnitude and the *FFT* of the same. Here the mean of the magnitude should work as a good discriminator between the activities since it is the length of the Euclidean vector the $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ forms in a 3-dimensional space, in other words the amplitude of the input signal. Also the mean frequency should give a good discrimination [31].

#### 3.4.4.2 Variance

Including the variance for the 24 values has been done in several previous work [1, 7, 29] among others. Like the mean, it is a good feature to have considering how much information about the data it provides.

#### 3.4.4.3 Eigenvalues of the covariance

Since the method for calculating the covariance, COV, does not work for a vector (gives only ones) it was decided to only use the $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ parts as a matrix in the calculation and exclude the magnitude. The result from COV itself was hard to use as features, but after some research it was discovered that principal component analysis (PCA) [24, 37] uses the eigenvalues of that result as part of its function and the decision was made to use the eigenvalues of the covariance as features.

#### 3.4.4.4 Max value

Max value is a common feature used in multiple past work [31, 13, 1, 7]. It should provide some interesting results which is why it was included in this analysis. Like the mean, the max value of the magnitude should provide a good discriminance between the activities. It can also be called the peak amplitude. The peak frequency should also show good discriminant values. [31]

#### 3.4.4.5 Energy

Calculating the energy of the input signal can be used to identify activities like walking, cycling and running with a single accelerometer [7], which makes it a promising feature to include in this thesis. The energy was calculated as sum of the squared coefficients divided by the size of the input, or sample window [7, 2], as below:

$$Energy = \frac{\sum x_i^2}{n} \tag{3.7}$$

where $x$ is the input vector and $n$ is the size of that vector. As stated before the energy is a feature often calculated using only the fourier transform of the input signal, but we wanted to see how the result could be affected if we were to use all 24 inputs in this calculation.

#### 3.4.4.6 Entropy

The use of entropy can help when discriminating input signals that have similar energy metrics but correspond to different activities such as cycling and running. For example, looking at data gathered with the device placed at the hip might give very similar energy values, but since cycling involves a uniform circular movement it can give high values in one dominant frequency coefficient but low for others, giving it a combined low value in entropy. Running might have a higher amount of coefficients with less difference between them compared to cycling but have values that provide a higher entropy value in total [2]. Entropy has been used together with energy and mean among other things in past work [7, 2]. Calculating the entropy of the input values were done using

$$Entropy = -\sum_{i=1}^{length(p)} (p_i(p_i > 0) \cdot \log p_i(p_i > 0)) \tag{3.8}$$

where $p$ comes from $[count, center] = hist(x, nbins)$[3] [21] and $p = count$ which was normalized as $p = \frac{p}{\sum p}$. Choosing a good value for $nbins$ took some thinking but finally ended up as the nearest integer rounded from the length of the inputs divided by 20 since we work with a large data set, $bins = round(\frac{length(X)}{20})$ [33]. Same as for the energy calculation the entropy was calculated using all 24 input values to see how the result would be affected, if calculating the energy and entropy on time domain signals would increase or decrease the accuracy and if they would provide features significant for the end result.

#### 3.4.4.7 Most recurrent values

From the hist-function [21] the three most recurrent values were also extracted. During the work with the hist-function, it became clear that there were multiple values that could count as the most recurrent, although, there was no clear single peak that could be extracted. Further testing showed that most of the scopes in the input files contained three values that occurred more times than others. However, having three most recurring values for each of the 16 *FFTs* in separate would be superfluous and pointless since they would not provide any relevant information about the behaviour of the data individually. In order to only have one value for each of the 16 *FFTs* the mean of the three values was used as a feature. Most recurrent value refers to the location of the bin centre on the x-axis from the hist-function.

#### 3.4.4.8 Zero Crossing Rate

The ZCR is one of the features that are normally not used for this type of data analysis, although it has been used to help distinguish walking from running by Lee and Mase [12] and to recognise stepping movements by Farringdon et al. [6]. The ZCR measures the rate at which a signal crosses a pre-set delimiter value [7], in this case set as 0 i.e. the ZCR in this case measures the rate which the signal goes from positive to negative and back [31]. Only the first four of the *FFT* values and the four *DCT*-values were used for the

[3]This functions is not recommended to use and one is supposed to use its successor Histogram [22] instead, but the differences between them [26] does not affect the result in this project in a negative way and the Histogram-function lacks the ability to get the location of each bin centre on the x-axis.

calculations of the Zero Crossing Rate. This since the $\hat{X}$, $\hat{Y}$, $\hat{Z}$ and magnitude, and the rest of the *FFT*s all had all, or next to all, values on the same side of zero (only positive or only negative values) they ended up as $NaN$ values that made the classification inconclusive.

#### 3.4.4.9 Total Harmonic Distortion

The Total Harmonic Distortion features are most used for audio signals, but they were included in this work as a test of what they might bring to the results and how much effect it would have. The THD-function does not work with imaginary inputs which meant that the *FFT* of the original values, numbers 5 to 8 in table 3.2, had to be excluded from the calculations. There was furthermore a slight problem when calculating the rest of the THD-values, many of them only gave $NaN$ as a result indicating that there were one or more calculations that gave -inf and when normalizing the value according to 3.6 the $NaN$-values were given and the whole classification became inconclusive. These problems were noticed quite early in the process and made the THD unusable for most of the values in table 3.2. However, when more data were collected and the problem with creating the scopes described in section 3.4.1 was solved, the problem with -inf- and resulting $NaN$-values disappeared and all values, except for the imaginary ones, could be used.

The use of the THD was not expected to have a huge impact in the classification since, like the ZCR, it was used with inputs not in its normal domain.

#### 3.4.4.10 Standard deviation

The standard deviation is also commonly used as a feature in this type of data analysis [7] but since the standard deviation is used when normalizing the features as 3.6 it was not a good idea to use as a feature in this master's thesis.

### 3.4.5 Feature selection

As different feature selection algorithms and filters mentioned in this thesis, such as CFS and ReliefF [1] or PCA [32, 14], might perform quite well, the decision was made to use a more naive approach to feature selection for this master's thesis work. The decision was made mainly based on the fact that most of the work that had to be done in Matlab was already done when setting up the analysis environment. The naive approaches used, consist of two methods where one of them is based on the result from the classification and the second from the classification method.

In the classification method used in this thesis work, the least squares method described in section 2.6, it is possible to extract the coefficients that has the biggest impact. It is the $\widetilde{W}$ in equations (2.5) and (2.7) that hold this information. This was described in section 2.6.1 and in section 2.6.2. The $\widetilde{W}$ was isolated and sorted in a descending order to get the features with the highest impact of the classification. This step was done after performing a classification using all of the 170 features. After sorting the values in $\widetilde{W}$ the classification was done again using first the two features with the highest impact and then using the three highest, four highest and so on up to using all of the features thus using the best subset of features all the way. The result from these classifications was plotted to get a visualisation of the performance when using the different number of features. The idea of using this method was derived through conversation with one of the advisors for the master's thesis and we also found that Ravi et al. [30] used this method in a smaller scale where it was easy to see what features had a big or a small impact on the result.

The other method to select the best subset of features was to first run the classification using only two features pair-wise. As opposed to the first method, this time every feature was paired with all the rest. This took some time to run since there were a lot of pairs to check. With 170 features and a complexity of $\binom{n}{k}$[4] where $n = 170$ and $k = 2$ there were just over of $14300$ numbers of classifications to run. The result was then put in a spreadsheet which was sorted in descending order depending on the percentage of correct classification. From this list the feature pairs with the highest percentage were chosen to the different subset sizes. An interesting thing to note here is that many of the pairs with a good result included much of the same features. The decision only to run this method using the features in pairs was made since it would take too long time to test subsets with higher number in the same way. To test all combinations of three features the complexity would be $\binom{170}{3}$ which will give about $800000$ numbers of classifications to run. Even this size would be a challenge and any more would be near impossible to calculate in reasonable time.

[4] This computes as $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

# 3.5 Classification

As mentioned in section 3.3, we used two different methods of classification. One where we used only two classes and one where we used all three classes. The differences from the background on least squares, was that we used -1 instead of 0 i the T-matrices (2.6). We did this to get a more defined difference between the classes.

## 3.5.1 Classification with three classes

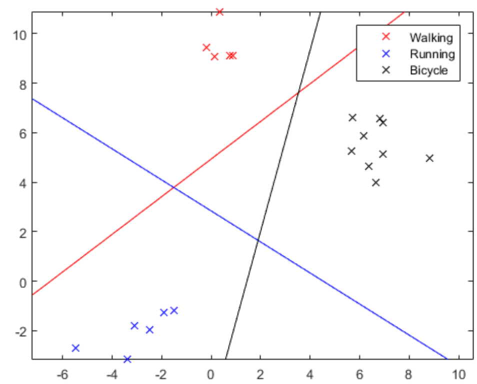


**Figure 3.4:** Classification with three classes, as can be seen a new point have the possibility to belong to none of the classes, which results in that this new point can't be classified.

As seen in the figure 3.4, which is the result of our implementation, the areas between the lines are different classes. In the case with three classes, there are 4 areas which have no class, and if a new point is placed in one of these areas, it will be impossible to classify this point. For this method the areas that do not belong to a class increases with N+1 where N are the number of classes.

For classification with three classes we utilised the method above, using least squares. We used the function from section 2.6 and more specifically subsection 2.6.2, where we used $k = 3$ and $n =$ *number of features*. This resulted in a vector *t* of length $k$, with the appearance from the result vector (3.1) in section 3.3. This classification will be called *Multiclass* for the remainder of this report.

### 3.5.2 Classification with two classes

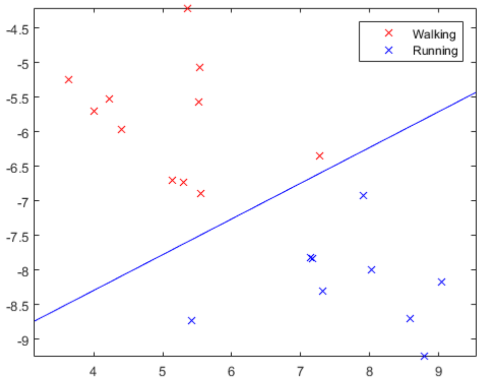


**Figure 3.5:** Classification with two classes, as can be seen a new point either belongs to one of the classes or the other.

As seen in the figure 3.5, classification with two classes, does not have the same problem as with classification with three classes, since there are no areas that do not belong to a class.

In the other method we used, we only matched 2 classes against each other at a time. We calculated one classification for each of the different pairs (Walk vs. Run, Walk vs. Bike and Run vs. Bike) and created a matrix with the following appearance:

$$\begin{pmatrix} \%Walking & \%Running & 0 & 0 \\ \%Walking & 0 & \%Bicycle & 0 \\ 0 & \%Running & \%Bicycle & 0 \end{pmatrix} \tag{3.9}$$

After this we calculated the mean of each column which resulted in a vector containing the percentage of instances of each class from all feature vectors, and the last column in the matrix contained percentages of inconsistency, although in this method this will always be 0, since all cases will be classifiable. The differences from classification with three classes, was that we used $k = 2$ for each of the three classifications, instead of one classification with $k = 3$, but everything else was the same. This classification will be called *Cross* henceforth.

We found that this classification was not optimal, since one of the classification has zero possibility to classify the data correctly. We then decided that if we could guess the most likely activity, it would be quite easy to increase the amount of correctly classified data. To do this, we used the Cross-classification and retrieved the index of the highest value of the result-vector, this index would correspond to the class the classification guessed was the most likely. This guess was used to eliminate the two-class classification, which did not contain the guessed activity, resulting in a matrix with the following appearance for the class walking:

$$\begin{pmatrix} \%Walking & \%Running & 0 & 0 \\ \%Walking & 0 & \%Bicycle & 0 \end{pmatrix} \tag{3.10}$$

and as for Cross-classification, we calculated the mean value of every column, and got a vector with the same appearance as Cross. This classification will henceforth be known as *Cross Guess*.

# Chapter 4

# Results

The results and findings from the work performed in this master's thesis is presented in this chapter. The features in the results are presented as numbers, which are defined in table D in chapter C in the appendix. Results not directly presented in this chapter but which still are important for the work can also be found in the appendix. Any acronyms used in this chapter can be seen in table A.1 in chapter A in the appendix.

## 4.1 Selection of the best subset of features

As seen in table 4.1, PWC55 is the best subset of features if one takes the percentages from classifications on each placement into account. PWM60 is the best if one only takes the percentages from the activities into account. PWC5B is the second best and WM70 is the third best in both cases.

From the results one can draw the following conclusion. The pairwise method with the our multiclass classification gives a better classification when you take placements into account and the pairwise method with the "cross guess"-classification gives a better classification when not taking the placements into account. Although, you get higher values when ignoring the placements. Our cross guess classification tops the table, with a quite high margin. And lastly the pairwise method seems to get better values than taking the values from $\widetilde{W}$. As one can see in table 4.1, the results from the balanced training data did not give as good classification as the unbalanced ones.

**Table 4.1:** This table 4.1, is a summary on the values from table C, table C, and table C. *AVG* is the average true positive classification over walking, running and bicycle. *Diff* is the difference between the highest and lowest value from the classifications on walking, running and bicycle. The tables are sorted according on first average (high to low) and then the diff (low to high), since we decided that a higher average is better than a low diff.

| **P** | | **AVG** | **Diff** |
|---|---|---|---|
| **PWM60** | ***CG*** | 81.03% | 17.32% |
| **PWC5 B** | ***CG*** | 72.79% | 27.11% |
| **WM70** | ***CG*** | 72.19% | 25.07% |
| **PWC55** | ***CG*** | 68.89% | 30.23% |
| **PWM75 B** | ***CG*** | 58.29% | 41.51% |
| **PWM60** | ***C*** | 54.02% | 11.55% |
| **PWC55** | ***C*** | 52.82% | 13.26% |
| **WM70** | ***C*** | 51.59% | 13.25% |
| **PWC5 B** | ***C*** | 48.52% | 18.08% |
| **PWM75 B** | ***C*** | 48.39% | 18.15% |
| **WM70** | ***M*** | 47.04% | 47.82% |
| **PWC95 B** | ***M*** | 42.32% | 55.27% |
| **PWC95 B** | ***CG*** | 41.35% | 58.45% |
| **PWC55** | ***M*** | 40.53% | 47.82% |
| **PWM60** | ***M*** | 39.13% | 55.80% |
| **PWM75 B** | ***M*** | 38.37% | 61.63% |
| **PWC5 B** | ***M*** | 37.72% | 59.21% |
| **WM50 B** | ***CG*** | 36.67% | 63.33% |
| **WM50 B** | ***C*** | 36.16% | 30.51% |
| **PWC95 B** | ***C*** | 35.96% | 30.58% |
| **WM50 B** | ***M*** | 34.36% | 65.24% |

| **NP** | | **AVG** | **Diff** |
|---|---|---|---|
| **PWC55** | ***CG*** | 90.68% | 8.64% |
| **PWC5 B** | ***CG*** | 90.48% | 12.91% |
| **WM70** | ***CG*** | 89.83% | 9.73% |
| **PWM60** | ***CG*** | 88.78% | 12.97% |
| **WM50 B** | ***CG*** | 81.17% | 12.80% |
| **PWC95 B** | ***CG*** | 80.96% | 21.75% |
| **PWM75 B** | ***CG*** | 79.59% | 19.36% |
| **PWC5 B** | ***M*** | 69.41% | 30.89% |
| **WM70** | ***M*** | 64.26% | 31.81% |
| **PWC55** | ***M*** | 62.45% | 37.37% |
| **PWC55** | ***C*** | 60.44% | 5.77% |
| **PWC5 B** | ***C*** | 60.32% | 8.61% |
| **PWM60** | ***M*** | 60.25% | 36.88% |
| **WM70** | ***C*** | 59.88% | 6.50% |
| **PWM60** | ***C*** | 59.19% | 8.65% |
| **WM50 B** | ***M*** | 55.93% | 33.94% |
| **PWM75 B** | ***M*** | 55.66% | 43.30% |
| **PWC95 B** | ***M*** | 54.30% | 39.73% |
| **WM50 B** | ***C*** | 54.11% | 8.53% |
| **PWC95 B** | ***C*** | 53.97% | 14.50% |
| **PWM75 B** | ***C*** | 53.06% | 12.90% |

## 4.2 Using all 170 features

Percentages of correct classification done with all of the 170 features extracted for each of the three activities. The result for each table is divided in placements and in the three way in which the classification was performed.

**Table 4.2:** Percentage of correct classification for walking with 170 features

| | All placements | Front pocket | Back pocket | Arm | Bag | Hand |
|---|---|---|---|---|---|---|
| **M** | 82.77 | 77.99 | 98.43 | 100 | 95.68 | 94.07 |
| **C** | 63.04 | 60.82 | 65.69 | 66.67 | 65.08 | 64.14 |
| **CG** | 94.56 | 91.23 | 98.54 | 100 | 96.22 | 97.62 |

**Table 4.3:** Percentage of correct classification for running with 170 features

| | All placements | Front pocket | Back pocket | Arm | Bag | Hand |
|---|---|---|---|---|---|---|
| **M** | 45.75 | 92.61 | 30.57 | 88.04 | 14.22 | 88.77 |
| **C** | 56.92 | 62.45 | 34.25 | 60.02 | 42.37 | 62.39 |
| **CG** | 85.38 | 93.67 | 34.81 | 90.03 | 63.56 | 93.58 |

**Table 4.4:** Percentage of correct classification for bicycle with 170 features

| | All placements | Front pocket | Back pocket | Arm | Bag | Hand |
|---|---|---|---|---|---|---|
| **M** | 67.49 | 88.60 | 93.17 | 95.83 | 92.94 | 79.52 |
| **C** | 60.45 | 63.01 | 62.73 | 62.5 | 64.78 | 63.45 |
| **CG** | 90.67 | 94.52 | 94.1 | 93.75 | 97.18 | 95.18 |

## 4.3 Features

The Figure 4.1 and 4.2 illustrate the performance of the different subsets of features from sizes of 2 up to 170 for the three classification methods. The features in the subsets are ranked, according to the two feature selection methods, from high impact to low. In figure 4.1 the features are selected with the pair-wise method and figure 4.2 illustrates the performance when the features are selected using the $\widetilde{W}$-matrix.

In both figures there are a clear moment when the accuracy of the classification get much better and then stays on that level. The increase is more prominent for *Cross Guess* and for *Multiclass* but also in *Cross* the increase in accuracy is noticeable. One can see in the two figures that increasing the number of features after this point does not yield much higher percentage to the accuracy.

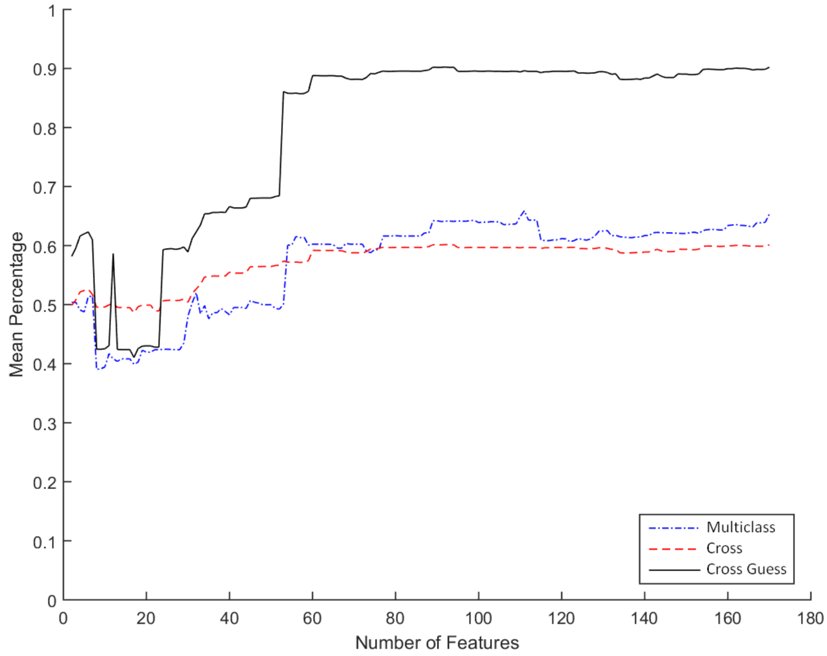


**Figure 4.1:** Figure showing the correctness in classification using an increasing subset of the best features from 2 to 170 where the features have been ranked according to their pair-wise multiclass classification

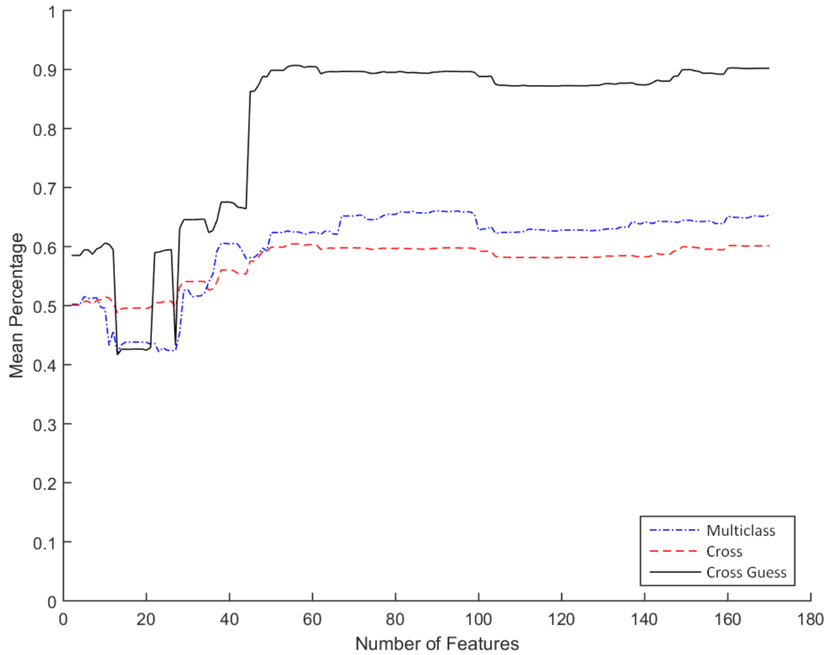


**Figure 4.2:** Figure showing the correctness in classification using an increasing subset of the best features from 2 to 170 where the features have been ranked according to their impact in the pair-wise cross guess classification. The ranking according to the Cross classification was the same as with the Cross Guess

As presented in the previous section, the *Cross Guess*-classification method, when using a subset of 60 features, is the one with the overall highest percentages of correct classification when using the same subset for both all placement combined as well as classifying the placements individually. The subset is presented in table 4.5. The subset of features that has the highest percentage when only classifying the placements together has 55 features and can be seen in table 4.6.

None of the radical increases in figures 4.1 and 4.2 corresponds exactly to the 55 and 60 features that were chosen as the best subsets but they are not very far from it. The percentage of correctness in classification with these subsets of features had a high value in correspondence to the number of features used which is why these numbers were chosen as the best subsets in the two cases.

**Table 4.5:** Table showing the 60 features in the subset that gave the best classification when the features were ranked from the Multiclass classification

| ***1-10*** | ***11-20*** | ***21-30*** | ***31-40*** | ***41-50*** | ***51-60*** |
|---|---|---|---|---|---|
| 4 | 20 | 77 | 126 | 55 | 118 |
| 12 | 32 | 89 | 53 | 59 | 124 |
| 78 | 86 | 97 | 54 | 63 | 37 |
| 11 | 57 | 19 | 125 | 2 | 138 |
| 35 | 61 | 41 | 70 | 72 | 7 |
| 3 | 65 | 85 | 69 | 166 | 74 |
| 16 | 15 | 23 | 141 | 170 | 123 |
| 90 | 31 | 44 | 43 | 56 | 168 |
| 98 | 38 | 142 | 51 | 60 | 100 |
| 73 | 50 | 129 | 52 | 64 | 99 |

**Table 4.6:** Table showing the 55 features in the subset that gave the best classification when the features were ranked from the Cross Guess classification

| ***1-10*** | ***11-20*** | ***21-30*** | ***31-40*** | ***41-50*** | ***51-55*** |
|---|---|---|---|---|---|
| 4 | 19 | 90 | 129 | 137 | 60 |
| 57 | 41 | 50 | 44 | 24 | 64 |
| 61 | 86 | 31 | 23 | 28 | 103 |
| 65 | 38 | 78 | 124 | 53 | 126 |
| 98 | 11 | 89 | 74 | 111 | 125 |
| 16 | 35 | 85 | 117 | 7 | |
| 73 | 15 | 2 | 37 | 70 | |
| 12 | 77 | 72 | 142 | 54 | |
| 20 | 97 | 138 | 40 | 56 | |
| 3 | 32 | 143 | 134 | 68 | |

## 4.4 Confusion Matrix

From table 4.7 one can see that in most cases, the hardest types to classify, are Bicycle and Running. For both, there is a quite high false negative rates, for Walking.

**Table 4.7:** Confusion Matrices over the classifications from the best subsets of features, including the one from Google, and the balanced and unbalanced versions of all features, to be used as reference points.

| | | Walking | Running | Bicycle | Ignored | NaN |
|---|---|---|---|---|---|---|
| **Google** | **Walking** | ***71%*** | 0% | 0% | 0% | 29% |
| | **Running** | 0% | ***55%*** | 0% | 0% | 45% |
| | **Bicycle** | 0% | 0% | ***24%*** | 44% | 32% |
| **170CG** | **Walking** | ***95%*** | 1% | 4% | 0% | 0% |
| | **Running** | 13% | ***85%*** | 2% | 0% | 0% |
| | **Bicycle** | 9% | 1% | ***91%*** | 0% | 0% |
| **PWC5BCG** | **Walking** | ***97%*** | 0% | 3% | 0% | 0% |
| | **Running** | 3% | ***91%*** | 7% | 0% | 0% |
| | **Bicycle** | 13% | 4% | ***84%*** | 0% | 0% |
| **PWC55CG** | **Walking** | ***96%*** | 1% | 3% | 0% | 0% |
| | **Running** | 9% | ***87%*** | 3% | 0% | 0% |
| | **Bicycle** | 11% | 0% | ***89%*** | 0% | 0% |
| **WM70CG** | **Walking** | ***95%*** | 1% | 4% | 0% | 0% |
| | **Running** | *13%* | ***85%*** | 2% | 0% | 0% |
| | **Bicycle** | 10% | *1%* | ***90%*** | 0% | 0% |
| **PWM60CG** | **Walking** | ***96%*** | 0% | *4%* | 0% | 0% |
| | **Running** | 10% | ***87%*** | 3% | 0% | 0% |
| | **Bicycle** | 17% | 0% | ***83%*** | 0% | 0% |
| **170BCG** | **Walking** | ***73%*** | 9% | 18% | 0% | 0% |
| | **Running** | 10% | ***78%*** | 12% | 0% | 0% |
| | **Bicycle** | 9% | 7% | ***84%*** | 0% | 0% |

**Table 4.8:** Confusion Matrices over the multiclass classification versions of the subsets from table 4.7.

| | | **Walking** | **Running** | **Bicycle** | **Inconclusive** |
|---|---|---|---|---|---|
| **170M** | **Walking** | ***83%*** | 1% | 6% | 10% |
| | **Running** | 19% | ***46%*** | 2% | 33% |
| | **Bicycle** | 24% | 0% | ***67%*** | 9% |
| **PWC5BM** | **Walking** | ***60%*** | 0% | 1% | 38% |
| | **Running** | 0% | ***89%*** | 3% | 7% |
| | **Bicycle** | 6% | 0% | ***58%*** | 35% |
| **WM70M** | **Walking** | ***82%*** | 1% | 5% | 11% |
| | **Running** | 20% | ***51%*** | 2% | 28% |
| | **Bicycle** | 30% | 0% | ***60%*** | 10% |
| **PWC55M** | **Walking** | ***82%*** | 0% | 6% | 12% |
| | **Running** | 14% | ***62%*** | 3% | 21% |
| | **Bicycle** | 43% | 0% | ***44%*** | 12% |
| **PWM60M** | **Walking** | ***79%*** | 0% | 6% | 14% |
| | **Running** | 16% | ***60%*** | 2% | 22% |
| | **Bicycle** | 48% | 0% | ***42%*** | 10% |
| **170BM** | **Walking** | ***38%*** | 18% | 14% | 31% |
| | **Running** | 15% | ***63%*** | 3% | 18% |
| | **Bicycle** | 11% | 10% | ***70%*** | 9% |

As can be seen in table 4.9, it is clear that, the unbalanced version of classification with all features, is better than the balanced one. It is also clear that it is not necessarily better to have more features, since PWC5B, which only uses 5 features has a higher mean value than "170", which uses 170 features. As mentioned earlier it is also very apparent that our cross guess classification method, is better than our implementation of the multiclass classification.

**Table 4.9:** Combined results of the true positives from the confusion matrices

| **CG** | **Google** | **170** | **PWC5B** | **PWC55** | **WM70** | **PWM60** | **170B** |
|---|---|---|---|---|---|---|---|
| **Walking** | 71% | 95% | 97% | 96% | 95% | 96% | 73% |
| **Running** | 55% | 85% | 91% | 87% | 85% | 87% | 78% |
| **Bicycle** | 24% | 91% | 84% | 89% | 90% | 83% | 84% |
| **Mean** | ***50%*** | ***90%*** | ***91%*** | ***91%*** | ***90%*** | ***89%*** | ***78%*** |

| **M** | **Google** | **170** | **PWC5B** | **WM70** | **PWC55** | **PWM60** | **170B** |
|---|---|---|---|---|---|---|---|
| **Walking** | 71% | 83% | 60% | 82% | 82% | 79% | 38% |
| **Running** | 55% | 46% | 89% | 51% | 62% | 60% | 63% |
| **Bicycle** | 24% | 67% | 58% | 60% | 44% | 42% | 70% |
| **Mean** | ***50%*** | ***65%*** | ***69%*** | ***64%*** | ***63%*** | ***60%*** | ***57%*** |

# Chapter 5
# Evaluation

In this chapter the results are evaluated and discussed, how the work with features and classification affected the outcome of the work and what might have been done differently in order to get a better result. At the end of this chapter, an assessment on whether the main goals for the thesis was met or not is presented.

## 5.1 Collecting the data

When looking at the way the data were collected there are some aspects that could have been done differently. Although the result did not match it, since the classification trained with unbalanced data had the highest overall performance, more focus on collecting a balanced amount of data is something which might produce better training data for the classifier and raise the accuracy when performing the classification [10]. The data used here had a much larger portion of walking data as to the other two activities and for the placements, the front pocket was used to a greater extent than the other four. For future research one would want to keep the data more balanced.

Also collecting more data would be preferable for future research. One interesting aspect to look at for the collection of data is how the classifier would perform if the training data had been collected in a different way. For this research it is collected almost an equal amount by objective subject, who collect and label the data themselves and the researchers involved in the project.

The data collected by the researchers will have some artificiality about it and it would be interesting to see what results could be generated using only data collected entirely by objective subjects. One also has to take into account the possibilities of mislabelling when the data is collected by uninvolved subjects. As mentioned in section 2.1 there is a higher probability of mislabelled data then the subjects are not supervised when collecting it. Even if precautions against this were taken during the implementation of the data collecting application, some of the data might be labelled as the wrong placement. This would have a great impact on the result and the outcome of the research. Looking at some of the percentages for some of the placements there are a few that had a very low accuracy in classification. It is impossible at this stage to determine if that accuracy is because of faulty training data entirely or if the fault is in the classification method. Therefore it would have been interesting to see what the result would be when comparing the results found in this thesis with the result of using more supervised labelled training data.

Both more data and more balanced data might have resulted in better results for e.g. the confusion matrix. The data collection application implemented for the thesis had some information that the result of the confusion matrix depends on. This means that all the data that were not collected using this application were discarded for this part and only used for the classifier, which only uses the accelerometer data in its calculations. The accelerometer sensor gives the same values regardless of what application it is used in, which is why the data that were collected not using the "correct" application could still be used in the classification. However, since the results from the collected data were compared to the results from the engine, i.e. Google's own activity recognition, data that were not collected with the application implemented for the project still had to be discarded in the end.

## 5.2 Confusion Matrix

As mentioned in section 4.4, the problem areas are Bicycle and Running, and both gives the highest false negative to Walking. It could be interesting to study the similarities between Walking and Running, and Walking and Bicycle, to see why it is harder to differentiate these than other combinations, e.g. Walking and Running. From this study one could find a set of features that may be better for these combinations.

## 5.3 Multiclass classification

As mentioned in section 3.5.1, there is a big problem with the our implementation of the multiclass classification, because of the areas that are impossible to classify. From the results of the confusion matrix, this is a problem even for our classification, since there are a lot of high percentages in the *inconclusive*-column in table 4.8. There are ways to minimize these areas, e.g. in figure 3.4 the areas between class areas could be joined with the class on the right or left, e.g. walking could be the area restricted by the red and blue lines instead of the red, blue and black lines [5]. This eliminates N areas for a classification of N classes, which reduces the areas to one area irrespective of the number of classes. Although even if these areas where completely removed, and the percentages of inconclusive classifications would be true positives instead, it would still in our case be an inferior classification compared to Cross Guess Classification. There is still one way to improve this method, instead of only checking if $s_i > 0$ in equation (2.9), one could check the largest value. This would result in an appeareance close to figure 5.1 instead of the appeareance of figure 3.4. In this figure, all inconclusive areas have disappeared and it would likely generate better results. The reason we did not use this method was that we completely missinterpreted the description of the method, which resulted in the version where only $s_i > 0$ was checked. Although, if one would add all of the inconclusive classification to the correct classification. The result would still have lesser values than the Cross Guess method, but the margins would almost be insignificant.

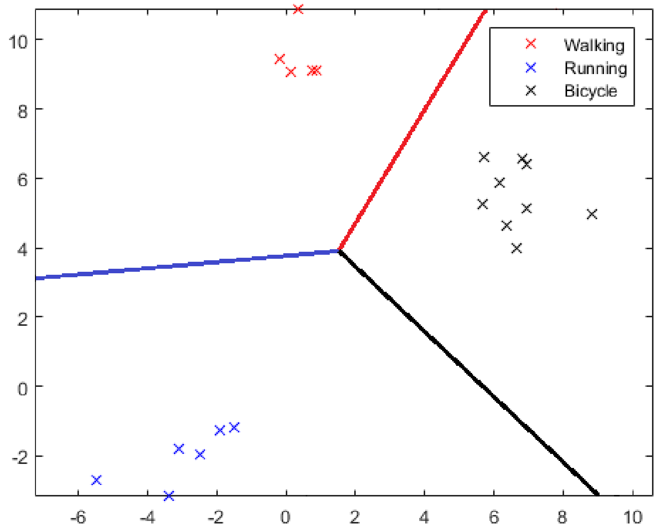


**Figure 5.1**

## 5.4 Cross and Cross Guess classification

The Cross classification was the worst of the three classification methods we tested. In most cases it barely reached 40% as one can see in table C, which is bad. Although the improvement of this classification, Cross Guess, was by far the best method in our study. But there are quite large risks with this method, since it is solely based on an ”educated” guess. The problem we see, is since it bases its guess on the Cross method, which is really bad it is likely that the guess is wrong. If the guess is wrong, this would make it unlikely for the classifier to give a correct answer, since one of the classifications, which the correct class is a part of, will be removed. Although, as can be seen in our result the Cross Guess method deliver very good results, which is interessting since it opens up the possibility to even better results if one bases the guess on a better evaluation method like e.g. Multiclass, and preferably the better variant mentioned earlier and not our implementation of it. But in either case it will most likely improve the results, giving a more accurate classification.

## 5.5 Other classification methods

The classification method used in this thesis is fairly simple and mainly uses the backslash-operator (Least Squares) in Matlab to perform the training of the Classification. Other methods which might be more complex and perform the classification in another fashion could generate a better result. There were, as mentioned, times when the least squares method had some difficulties differentiating between classes. Other more elaborate methods might not have the same problem. On the other hand the simplicity in least squares also has the advantage that it is easy to test for errors and when the main focus of the research is not in the classification method itself it still provides sufficient performance accuracy to the research.

## 5.6 Choice of activities

In the beginning of our study, we decided that we would concentrate on the three activities, walking, running, and bicycle. We chose these because it would be quite easy to collect data on these, and because these are the activities most persons would value in an activity tracking application. During the project we have had a thought on how more activities would affect our classifications, since there would be a higher probability to classify an activity falsely since there would be more cases to take into account. It is likely that all of our classifications would fail or at least be less accurate. Cross would likely have an even worse classification since more classes would make the correct one a little less significant. This would also affect the Cross Guess, since the initial guess probably would becompletely wrong. For our multiclass the area of the inconclusive areas, would be greater since the amount of areas would increase.

As mentioned above there are ways to improve the methods, and we think that e.g. cross guess, based on the result from multiclass, could handle more classes. There are also ways of modifying the cross guess, e.g. if the number of classes is greater than three, it is possible to build the matrix 5.1 and reduce it to matrix 5.2 from classifications with three classes instead of two, and for the general case where $Classes > N$, classifications with $N - 1$ classes. For the case with four classes, the matrix would have the following appearance:

$$\begin{pmatrix} \%C1 & \%C2 & C3 & 0 \\ \%C1 & \%C2 & 0 & \%C4 \\ \%C1 & 0 & \%C3 & \%C4 \\ \%0 & \%C2 & C3 & \%C4 \end{pmatrix} \tag{5.1}$$

And for the correct class $C1$ reduce it to:

$$\begin{pmatrix} \%C1 & \%C2 & C3 & 0 \\ \%C1 & \%C2 & 0 & \%C4 \\ \%C1 & 0 & \%C3 & \%C4 \end{pmatrix} \tag{5.2}$$

by removing, in this case, the last line.

## 5.7 Placements

As one can see in both the 4.2 and in C.4, C.5 and C.6 in appendix C some of the placements have a higher percentage than the result for all the other placements combined. This is an important aspect to consider when using the results from this thesis in industry. The subset of 60 features with the best average percentage was chosen solely on the percentages of all the placements, i.e. when using all data in the selection process, and there are certainly other subsets of features that will have a higher average when looking at the placements individually.

In order to obtain as accurate a classification as possible one can use different feature sets depending on the placement a person uses for the device when performing the activity. However, this will require more storage for the different sets and an initial guess of the placement would be necessary. Since it is the activity and not the placement that are to be recognised it would be a possibility to use other sensors or methods to find the location of the device and then perform the classification accordingly.

# 5.8 Features

When looking at the two tables 4.5 and 4.6, one can see that there are many features that appear in both, even if not ranked the same internally with some exceptions like the mean of the magnitude which sits at the top of both tables. This shows that these features have a good synergy, or correlation, between them and create distinct clusters both in the 55- and 60-dimensional space they appear in. Something that is very interesting, and given more time would be preferable to further investigate, is the fact that the results can vary a lot depending on as little as one single feature as seen in figures 4.1 and B.1. A thought on why the difference is so palpable is that when adding a new feature the $n$-dimensional space created is created in such a way that the $n$-dimensional points alters the decision boundary. This can be seen already when moving from two dimensions to three. When adding a $Z$ input to a point consisting of $(X, Y)$ in a plane, the location of that point in the sphere moves and the two-dimensional decision boundary line transforms into a decision boundary plane. For higher dimensions this plane becomes a figure with different shape for each new feature. For each added feature the dimension goes from $\mathbb{R}^n$ to $\mathbb{R}^{n+1}$.

Studying the features in the two tables further, one can see that there are many which are calculated from the $\hat{Z}$ input and from the magnitude, with the latter being represented the most. To see that the $\hat{Z}$ input is important leads to the thought that the activities studied in this thesis are quite similar in their movements in the $X$- and $Y$-axis while the $Z$-axis is the one providing the distinction in the movement. The fact that the magnitude is even more represented among the best features is not surprising since it is calculated from the three axes and follows the appearance of the axis. If one of the axes has a higher value for one activity while the others remain small in difference, the magnitude will behave accordingly and will thus always affect the classification and be an important feature to calculate.

## 5.8.1 Orientation of the inputs

The decision to remove the significance of the orientation of the input values is one aspect that might have had a big impact on the result for this research. Since it was not featured in any of the related work studied it was somewhat unchartered territory on how it would affect the calculations and the classification. The possibility has already been mentioned for the Zero Crossing Rate in the section above but since there was no extensive testing on the matter one can only speculate about how it influenced the calculations of other features. The fact that the magnitude showed such prominent result in terms of belonging to the two best subsets of features also spurs the thoughts on how much impact this part of the work really had, but one can also argue that without removing the significance of the orientation, in order to maintain the same level of equality between the inputs, only the magnitude should be used as an input. The positive aspects of making the orientation irrelevant is that no thoughts or cautions have to be taken when collecting the data regarding on how the device is placed, the inputs are equal and a $Y$ input does not have the same value as an $X$ input when rotated 90 degrees, it can still be seen as a $Y$ input. Looking at this more in the aspect of collecting data a subject in an out-of-lab setting is allowed to be even more real in its movements if they do not have to care about the device being in the correct orientation all the time.

Overall we feel that the decision to make the orientation irrelevant was the right one but it would, of course, be interesting to test this aspect and to see whether removing the significance in orientation does have a positive or a negative effect. Unfortunately it would take too long time to do for this thesis since it will mean performing the whole feature calculation and classification again and to compare the results with the one already generated.

To further improve the result from this step one can add a weight when normalizing the rows, i.e. when producing the $\hat{U}$-vector in equation (3.4), in order to get a better focus on the row in question. It might not change the outcome in a big manner but it might provide a bit more distinction between some inputs.

### 5.8.2 Feature extraction

In total a number of 24 initial values were extracted from the three axes $X$, $Y$ and $Z$ and most of them were used in all the 9 different feature categories. Except for the Zero Crossing Rate all of the feature categories are represented in the two tables 4.5 and 4.6, with some more prominent than others. The mean, variance and max value calculations, as mentioned before in section 3.4.4, are all frequently represented in the tables proving that they are good features when classifying accelerometer data. Looking at the mean values they had 50% representation in both the tables and since they were only calculated for 8 of the 24 initial values it would probably have been a good idea to perform these calculations on more values. It would have been interesting to see how the mean would perform for the rest of the values, if they would have replaced some of the other categories and made them less prominent and how it would have affected the size of the subsets, especially since 25% of the mean features are found among the six and eight top features in the two tables.

When extracting features using the eigenvalues of the covariance the magnitude was not included as stated in section 3.4.4.3. That decision was made since it, at the time, made more sense to only use the three axes as a 3-by-3 matrix instead of as individual arrays. However, since the magnitude has shown very promising results for other feature categories, this decision might potentially have had a big impact in the result. As for the mean values, it would have been interesting to see how this feature would have performed and affected the result.

Regarding the impact of energy and entropy features calculated with values from another domain than usual, they were very low when considering the two subsets presented in the results. There were only three such features in each of the two tables, all of which were at the end of table 4.5. This was an expected outcome but it would have been an interesting aspect to the field if the results had shown the opposite.

The energy and entropy were overall not represented much in the two tables in the results, in total there where less than 20% of features based on these feature categories with the entropy only being represented in three of them. Past work such as Figo et al. [7] has shown that the calculation of entropy is at a high cost in computation which means that it is not really suited for use in a mobile device. the Energy is at medium cost while others like mean, max value, ZCR are very low in terms of computational cost and is greatly suitable for use in a mobile device. With that in mind the fact that the entropy did not have a big impact on the results was positive. During this research the classification was not performed without the features of entropy, but one can conclude that removing the entropy for further research will not have a negative effect on the classification and is hence still useable for a device which have limitations in power consumption and computational performance. However, devices such as cell phones are becoming more and more powerful with higher processing performance and better battery capacity and the aspect of limitations might be less important in the future.

Another feature category that had an interesting result considering our expectations was Zero Crossing Rate, which had problems when calculating it in its normal domain. The fact that the $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ values did not contribute anything in this feature were interesting since ZCR is usually used in the time domain [31] but here they only provided $NaN$-values when using inputs from that domain. The values that worked for ZCR were some of the inputs calculated from the *FFT* and the values from the *DCT* seen in table 3.2. One explanation for this anomaly might be that the significance of the orientation for the original inputs was removed which created a disturbance when calculating the ZCR. The most obvious inputs not to work with ZCR are numbers 13 to 16 in table 3.2, the absolute value of the *FFT*, since they are made to only have positive values. As mentioned in section 3.4.4.8, the Zero Crossing Rate feature have been used to help distinguish walking from running [12] and to recognise stepping movements [6] in the past although that is not its usual field, but as seen in tables 4.5 and 4.6 none of the subset that gave the best result include any of the features from the Zero Crossing Rate calculations. This also might be an effect of the orientations of the inputs and one might argue that removing the significance of the orientation for the data is not the best approach in this case.

### 5.8.3 Feature selection

The way that feature selection is performed in this master's thesis is, as mentioned in section 3.4.5, done with two naive methods. At the time when the decision was made to find the best subset of features this seemed as a promising method in how to get a good result, it was fairly easy to set up and the results were easy to interpret and to use. Afterwards, however, the time that it took to run the selection made it clear that a selection method based on an algorithm might have been more preferable to use, especially if it would have reduced the run time. Even if the setup used was fairly easy, time that could have been saved by using a faster method could have been spent to widen the scope of the analysis or to go even deeper in some parts of the analysis.

With the results in hand, one can clearly see the potential that this field has and that even if this research only covers a narrow part it still has shown that when analysing data the features play a very important part for the result.

## 5.9 Errors

When creating the feature matrices and feature vector in matlab the program crashed when processing some of the files. The source of the crash was found to be that when creating the scopes for each file the amount of rows of data in that file made some scopes become very small, some as small as only five rows of data. This created a fault when performing the calculations for equalizing the orientation on the inputs. This is a problem that should not arise here since those calculations are only performed on one row at the time except for the step when the $\hat{U}$ is created. In that step the calculations use at most seven rows, three before and three after the row that is being calculated, but it should not generate a fault if there are less than seven rows since the implementation of the calculations have special cases for the first and last rows. To solve the problem the calculations of scopes were changed to ignore any scope with less than 10 entries in it. A normal sized scope has approximately 1000 entries and a scope with only 10 entries will not have a significant effect on the result.

Some of the features were found to have several values of $-\mathrm{inf}$ in the beginning of the feature extraction part of the analysis which in turn resulted in $\mathrm{NaN}$-values when normalising the features using the standard deviation and the classification gave only inconclusive results. The fault originated from the fact that the accelerometer sensor sometimes stopped producing data which result in gaps in the timestamps. To solve this problem those scopes containing such a gap were excluded and features were not calculated from them. This resulted in some loss of data but since there were not more than 30 scopes containing gaps in total out of several thousands, it did not have a substantial effect on the result.

In several previous sections in this report the significance of the amount of data and whether the data should be balanced or not has been discussed. As one can see in the result this research show better numbers for non-balanced data than for balanced data which was not an outcome that was expected. One thing that we have struggled with during the work and which might be the reason for this unexpected result is that not enough data were collected. The initial aim for the data collection part was at least around 4 or 5 hours of data for each activity and placement and this goal was not met. When analysing data, having data to analyse is crucial and the fact that we did not manage to collect as much data as intended might have had a big impact on the results.

One of our most significant errors, was that we totally missinterpreted the description of multiclass classification. This most likely have affected the result from this method, generating worse classifications. Although, as mentioned in section 5.3, even if all the inconclusive areas would be classified, it would at most get close to our cross guess method, but not surpass it. But it is not possible to make a generalization based on this data alone. It will most likely also affect the subset of features for those variants which used the multiclass method. But it will not affect the main goal of the master's thesis. Because, using the correct version of the multiclass method would only result in better classifications, which in turn would make our claim, that it is possible to get good classifications when only using the accelerometer data, more legitimate.

## 5.10 Verdict

By applying the methods used in this study one can conclude that it is possible to get a satisfactory classification when only taking data from the accelerometer sensor in a cell phone into account. As stated in the evaluation chapter there are many possible improvements that may raise the accuracy for the classification even higher.

We have also concluded that it is possible to isolate single placements for the device and still get very good results. Further conclusions have been made that whether one uses the placements separately or in combination the best possible subset of features may vary.

When looking at both the highest average in percentages and the smallest difference between the largest and smallest value, the features from PWC55, shown in table 4.6, is the best subset of features. Although, this does not take separate placements into account and only looks at the activities in full. When the seperate placements are taken into account, the features from PWM60, in table 4.5, are the best ones.

Furthermore, when looking at the diffrence between the best subsets for placements versus non-placements, the average accuracy drops significantly (close to 10 percentage) and the difference between highest and lowest value approximately doubles. These obervations lead to the simple fact that there are some factors that have to be taken into account for selecting the best subset of features.

This study has only been done for the three activities walking, running and bicycle and when using the two best subsets of features mentioned above, the areas with most false negatives, i.e most incorrect classifications, are walking and running when running is the correct activity, and walking and bicycle when bicycle is the correct activity.

A key observation in our study is the fact that when combining different evaluation methods one might obtain a better result.

# Chapter 6
# Future Work

As mentioned in chapter 2 the field of data analysis is vast with many varieties on how to approach it. The results presented in the master's thesis has been an early step in the process of using only the accelerometer in a cell phone for accurate activity recognition and there are many ways in which one might continue this work and further develop the research. In the time that this thesis has been performed several possibilities in improvements, future work and development have been thought of. Some of them are specified in this section.

## Features

We used nine different categories of features for this thesis but the number of possible features to extract is almost as vast as the field of data analysis itself. There are many more categories one can calculate to get other features for a future project. Examples of such feature categories are RMS, correlation and cross-correlation, the range of the input, DC component and many more. Also one can use the integration of a signal as a feature, which could be very interesting. Some of these we tried to use for our work but we either were not able to implement it correctly of did not have to time to look into it further. Correlation is such a feature. When reading about correlation we figured that it might be a good feature to have. Previous work such as [2] uses correlation as it can recognise movements of multiple body parts. The drawback is that it is calculated using only two input vectors and since we work with three, we initially did not think we could have any use for it. It was later in the project, when we unfortunately did not have the time to pursue it, that we realized that one could calculate the correlation for three axes in a pairwise fashion i.e. $(x, y), (x, z)$ and $(y, z)$ and then select the pair that had the largest coefficients to discriminate between activities [7].

## Classifying

One aspect of this thesis we would have looked into if we had more time is the use of other classifying methods. One of our sub goals were to use a simple classifying method like least squares in order to get a result and a baseline and then compare our findings when using the same subset of features in other classifying methods. Some of the methods that we would have wanted to compare were Naive Bayes and SVM as they have been used in many of the previous work that we studied. Also linear discriminant analysis (LDA) would have been interesting to try. The idea of using multiple classification methods were based on the feature interest we had in this thesis. Since our work was heavily focused on features, we wanted to see if the same feature subset that displayed high results for one classifying method would show the same high result when using another. Other classifying methods might be faster in classifying the activity and might be a better choice for mobile devices than least squares but does not utilize the same features, this would have a big impact when selecting the final way in using the accelerometer data to recognise activities in a satisfying way in a mobile phone.

### Parallelize workload

For this thesis we only tested pairwise feature performance because of the time it would take to test every triples when having as many features as we have had. One way to reduce the run time and to be able to test sets of three, four, five or even more is to parallelize the calculations. The implementation we have created can for future work be divided into multiple computers and cores to parallelize the workload and to make the testing of larger sets possible, as it is not in its current configuration.

# Appendices

# Appendix A
# Acronyms

**Table A.1:** Acronyms

| | |
|---|---|
| ***M*** | *Multiclass classification in section* 3.5.1 |
| ***C*** | *Cross classification as in section* 3.5.2 |
| ***CG*** | *Cross Guess classification as in section* 3.5.2 |
| ***AP*** | *All placements* |
| ***FP*** | *Front pocket* |
| ***BP*** | *Back pocket* |
| ***PWM*** | *Pair-Wise M, M defined above* |
| ***PWC*** | *Pair-Wise CG, CG defined above* |
| ***WM*** | *W-tilde ($\widetilde{W}$) M, M defined above*[1] |
| ***P*** | Results from the classification on each placement is taken into account. |
| ***NP*** | When only the result from the whole activity is taken into account, and the classifications from placements was ignored. |
| ***B*** | If the method is based on a balanced Feature matrix |
| ***MAG*** | *referres to the combination of $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ according to equation* (3.2) |
| ***MRV*** | Most recurrent value |

# Appendix B

# Pictures

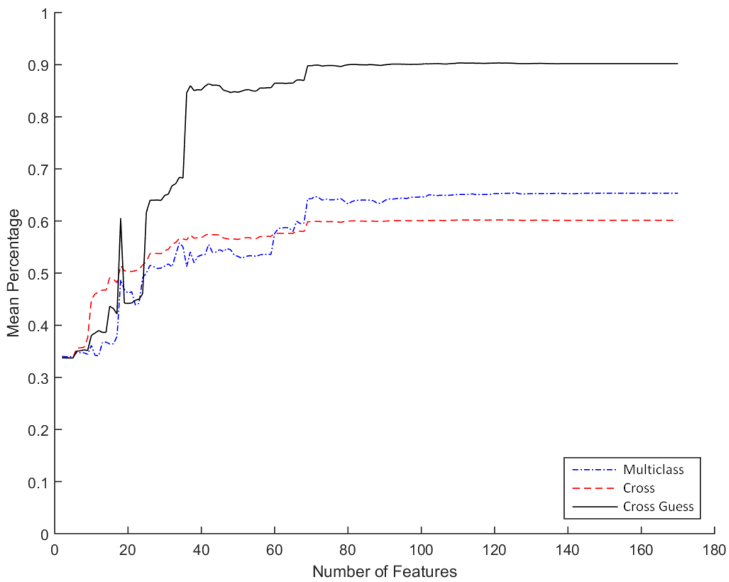


**Figure B.1:** Figure showing the correctness in classification using an increasing subset of the best features from 2 to 170 where the features have been ranked according to their impact in the $\widetilde{W}$-vector

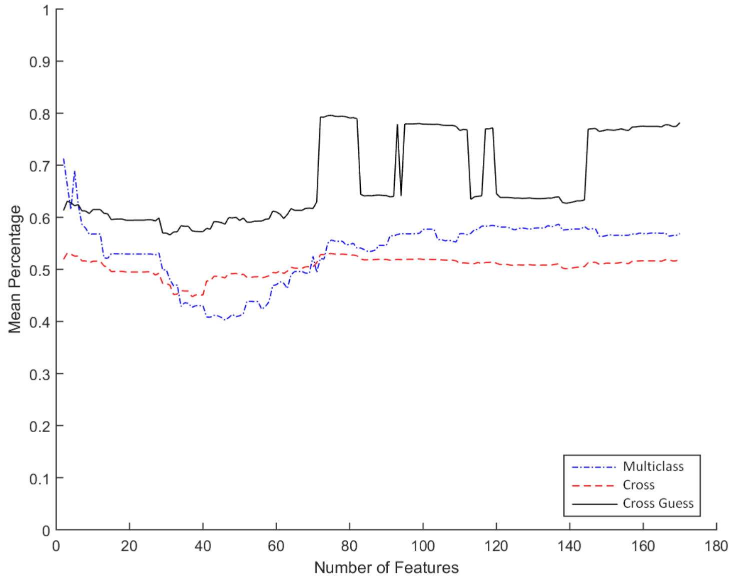


**Figure B.2:** Figure showing the correctness in classification using an increasing subset of the best features from 2 to 170 where the features have been ranked according to their pair-wise classification with balanced training data

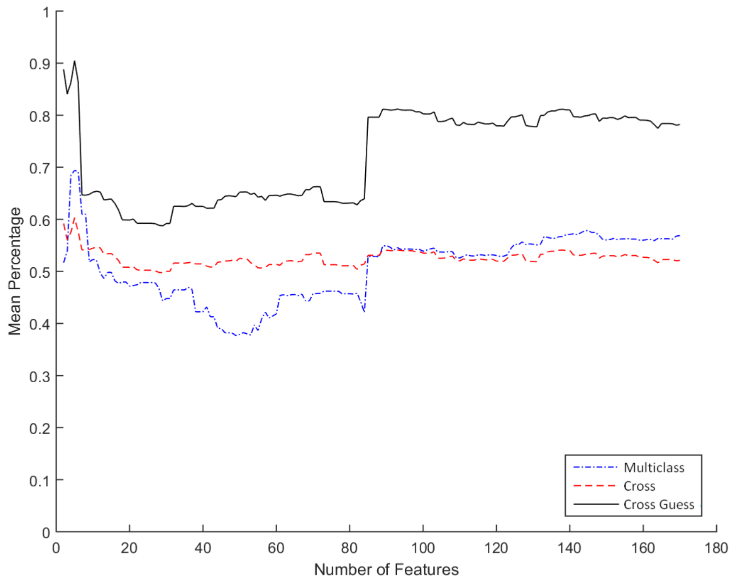


**Figure B.3:** Figure showing the correctness in classification using an increasing subset of the best features from 2 to 170 where the features have been ranked according to their pair-wise classification with balanced training data

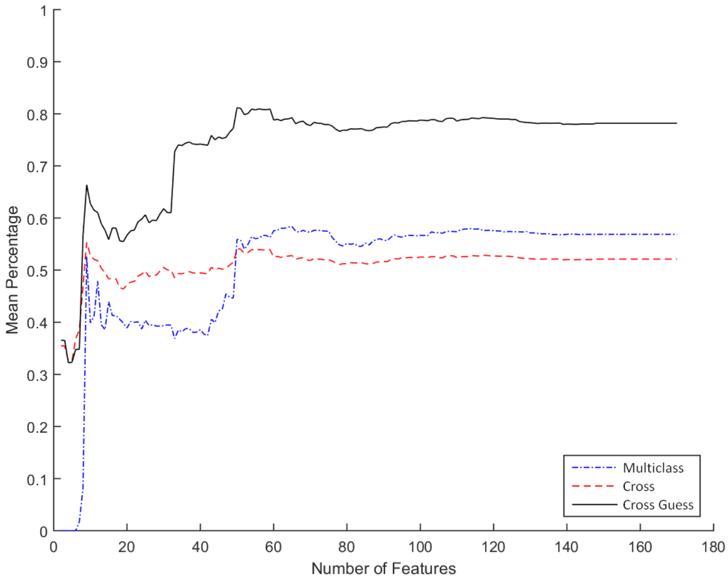


**Figure B.4:** Figure showing the correctness in classification using an increasing subset of the best features from 2 to 170 where the features have been ranked according to their impact in the $\widetilde{W}$-vector with balanced training data

# Appendix C

# Tables

**Table C.1:** Calculations on the values from C.4,C.5, and C.6 on M. *Max** for *P*, it was Avg(Max) for *NP* it was only Max, same for *Min**.

| Multiclass | | P | | NP |
|---|---|---|---|---|
| | | **min** | **max** | **value** |
| **PWM75 B** | **Walking (W)** | 28.35% | 100.00% | 28.35% |
| | **Running (R)** | 15.11% | 100.00% | 66.97% |
| | **Bicycle (B)** | 71.65% | 100.00% | 71.65% |
| | **Min(W,R,B)** | 15.11% | 100.00% | 28.35% |
| | **Max(W,R,B)** | 71.65% | 100.00% | 71.65% |
| | **Avg(W,R,B)** | ***38.37%*** | 100.00% | ***55.66%*** |
| | **Diff(Min*,Max*)** | ***61.63%*** | | ***43.30%*** |

Continued on next page

**Table C.1 – continued from previous page**

| Multiclass | | P | | NP |
|---|---|---|---|---|
| | | **min** | **max** | **value** |
| **PWC5 B** | **Walking (W)** | 18.28% | 100.00% | 60.47% |
| | **Running (R)** | 48.44% | 97.92% | 89.33% |
| | **Bicycle (B)** | 46.45% | 92.89% | 58.44% |
| | **Min(W,R,B)** | 18.28% | 92.89% | 58.44% |
| | **Max(W,R,B)** | 48.44% | 100.00% | 89.33% |
| | **Avg(W,R,B)** | ***37.72%*** | 96.94% | ***69.41%*** |
| | **Diff(Min*,Max*)** | ***59.21%*** | | ***30.89%*** |
| **PWC95 B** | **Walking (W)** | 29.17% | 100.00% | 29.17% |
| | **Running (R)** | 28.89% | 93.98% | 64.84% |
| | **Bicycle (B)** | 68.90% | 98.80% | 68.90% |
| | **Min(W,R,B)** | 28.89% | 93.98% | 29.17% |
| | **Max(W,R,B)** | 68.90% | 100.00% | 68.90% |
| | **Avg(W,R,B)** | ***42.32%*** | 97.59% | ***54.30%*** |
| | **Diff(Min*,Max*)** | ***55.27%*** | | ***39.73%*** |
| **WM50 B** | **Walking (W)** | 34.76% | 100.00% | 34.76% |
| | **Running (R)** | 4.00% | 100.00% | 68.70% |
| | **Bicycle (B)** | 64.33% | 98.80% | 64.33% |
| | **Min(W,R,B)** | 4.00% | 98.80% | 34.76% |
| | **Max(W,R,B)** | 64.33% | 100.00% | 68.70% |
| | **Avg(W,R,B)** | ***34.36%*** | 99.60% | ***55.93%*** |
| | **Diff(Min*,Max*)** | ***65.24%*** | | ***33.94%*** |

Continued on next page

**Table C.1 – continued from previous page**

| Multiclass | | P | | NP |
|---|---|---|---|---|
| | | **min** | **max** | **value** |
| **PWM60** | **Walking (W)** | 74.36% | 98.43% | 79.01% |
| | **Running (R)** | 0.89% | 86.36% | 59.60% |
| | **Bicycle (B)** | 42.13% | 100.00% | 42.13% |
| | **Min(W,R,B)** | 0.89% | 86.36% | 42.13% |
| | **Max(W,R,B)** | 74.36% | 100.00% | 79.01% |
| | **Avg(W,R,B)** | ***39.13%*** | 94.93% | ***60.25%*** |
| | **Diff(Min*,Max*)** | ***55.80%*** | | ***36.88%*** |
| **PWC55** | **Walking (W)** | 73.10% | 99.22% | 81.43% |
| | **Running (R)** | 4.44% | 93.85% | 61.87% |
| | **Bicycle (B)** | 44.06% | 100.00% | 44.06% |
| | **Min(W,R,B)** | 4.44% | 93.85% | 44.06% |
| | **Max(W,R,B)** | 73.10% | 100.00% | 81.43% |
| | **Avg(W,R,B)** | ***40.53%*** | 97.69% | ***62.45%*** |
| | **Diff(Min*,Max*)** | ***47.82%*** | | ***37.37%*** |
| **WM70** | **Walking (W)** | 80.82% | 100.00% | 82.36% |
| | **Running (R)** | 0.44% | 92.03% | 50.55% |
| | **Bicycle (B)** | 59.87% | 92.55% | 59.87% |
| | **Min(W,R,B)** | 0.44% | 92.03% | 50.55% |
| | **Max(W,R,B)** | 80.82% | 100.00% | 82.36% |
| | **Avg(W,R,B)** | ***47.04%*** | 94.86% | ***64.26%*** |
| | **Diff(Min*,Max*)** | ***47.82%*** | | ***31.81%*** |

**Table C.2:** Calculations on the values from C.4,C.5, and C.6 on C. *Max** for *P*, it was Avg(Max) for *NP* it was only Max, same for *Min**.

| **Cross** | | **P** | | **NP** |
|---|---|---|---|---|
| | | **min** | **max** | **value** |
| **PWM75 B** | **Walking (W)** | 44.65% | 66.67% | 44.65% |
| | **Running (R)** | 42.96% | 66.67% | 56.98% |
| | **Bicycle (B)** | 57.55% | 66.27% | 57.55% |
| | **Min(W,R,B)** | 42.96% | 66.27% | 44.65% |
| | **Max(W,R,B)** | 57.55% | 66.67% | 57.55% |
| | **Avg(W,R,B)** | ***48.39%*** | 66.54% | ***53.06%*** |
| | **Diff(Min*,Max*)** | ***18.15%*** | | ***12.90%*** |
| **PWC5 B** | **Walking (W)** | 37.20% | 66.67% | 64.50% |
| | **Running (R)** | 52.47% | 66.67% | 60.57% |
| | **Bicycle (B)** | 55.89% | 66.46% | 55.89% |
| | **Min(W,R,B)** | 37.20% | 66.46% | 55.89% |
| | **Max(W,R,B)** | 55.89% | 66.67% | 64.50% |
| | **Avg(W,R,B)** | ***48.52%*** | 66.60% | ***60.32%*** |
| | **Diff(Min*,Max*)** | ***18.08%*** | | ***8.61%*** |
| **PWC95 B** | **Walking (W)** | 44.68% | 66.67% | 44.68% |
| | **Running (R)** | 25.69% | 66.67% | 59.18% |
| | **Bicycle (B)** | 37.50% | 66.27% | 58.06% |
| | **Min(W,R,B)** | 25.69% | 66.27% | 44.68% |
| | **Max(W,R,B)** | 44.68% | 66.67% | 59.18% |
| | **Avg(W,R,B)** | ***35.96%*** | 66.54% | ***53.97%*** |
| | **Diff(Min*,Max*)** | ***30.58%*** | | ***14.50%*** |

Continued on next page

**Table C.2 – continued from previous page**

| C | | P | | NP |
|---|---|---|---|---|
| | | **min** | **max** | **value** |
| **WM50 B** | **Walking (W)** | 43.66% | 66.67% | 49.09% |
| | **Running (R)** | 9.19% | 66.67% | 57.62% |
| | **Bicycle (B)** | 55.62% | 66.67% | 55.62% |
| | **Min(W,R,B)** | 9.19% | 66.67% | 49.09% |
| | **Max(W,R,B)** | 55.62% | 66.67% | 57.62% |
| | **Avg(W,R,B)** | ***36.16%*** | 66.67% | ***54.11%*** |
| | **Diff(Min*,Max*)** | ***30.51%*** | | ***8.53%*** |
| **PWM60** | **Walking (W)** | 63.48% | 66.67% | 64.11% |
| | **Running (R)** | 43.13% | 63.37% | 58.00% |
| | **Bicycle (B)** | 55.46% | 66.67% | 55.46% |
| | **Min(W,R,B)** | 43.13% | 63.37% | 55.46% |
| | **Max(W,R,B)** | 63.48% | 66.67% | 64.11% |
| | **Avg(W,R,B)** | ***54.02%*** | 65.57% | ***59.19%*** |
| | **Diff(Min*,Max*)** | ***11.55%*** | | ***8.65%*** |
| **PWC55** | **Walking (W)** | 62.93% | 66.67% | 63.90% |
| | **Running (R)** | 36.23% | 65.60% | 58.13% |
| | **Bicycle (B)** | 59.30% | 65.97% | 59.30% |
| | **Min(W,R,B)** | 36.23% | 65.60% | 58.13% |
| | **Max(W,R,B)** | 62.93% | 66.67% | 63.90% |
| | **Avg(W,R,B)** | ***52.82%*** | 66.08% | ***60.44%*** |
| | **Diff(Min*,Max*)** | ***13.26%*** | | ***5.77%*** |

Continued on next page

**Table C.2 – continued from previous page**

| C | | P | | NP |
|---|---|---|---|---|
| | | min | max | value |
| **WM70** | **Walking (W)** | 61.40% | 66.67% | 63.15% |
| | **Running (R)** | 33.53% | 63.19% | 56.65% |
| | **Bicycle (B)** | 59.83% | 64.66% | 59.83% |
| | **Min(W,R,B)** | 33.53% | 63.19% | 56.65% |
| | **Max(W,R,B)** | 61.40% | 66.67% | 63.15% |
| | **Avg(W,R,B)** | ***51.59%*** | 64.84% | ***59.88%*** |
| | **Diff(Min*,Max*)** | ***13.25%*** | | ***6.50%*** |

**Table C.3:** Calculations on the values from C.4,C.5, and C.6 on CG. *Max** for *P*, it was Avg(Max) for *NP* it was only Max, same for *Min**.

| Cross Guess | | P | | NP |
|---|---|---|---|---|
| | | min | max | value |
| **PWM75 B** | **Walking (W)** | 24.10% | 100.00% | 66.97% |
| | **Running (R)** | 64.44% | 100.00% | 85.47% |
| | **Bicycle (B)** | 86.33% | 99.40% | 86.33% |
| | **Min(W,R,B)** | 24.10% | 99.40% | 66.97% |
| | **Max(W,R,B)** | 86.33% | 100.00% | 86.33% |
| | **Avg(W,R,B)** | ***58.29%*** | 99.80% | ***79.59%*** |
| | **Diff(Min*,Max*)** | ***41.51%*** | | ***19.36%*** |

Continued on next page

**Table C.3 – continued from previous page**

| Cross Guess | | P | | NP |
|---|---|---|---|---|
| | | **min** | **max** | **value** |
| **PWC5 B** | **Walking (W)** | 55.81% | 100.00% | 96.75% |
| | **Running (R)** | 78.71% | 100.00% | 90.85% |
| | **Bicycle (B)** | 83.84% | 99.69% | 83.84% |
| | **Min(W,R,B)** | 55.81% | 99.69% | 83.84% |
| | **Max(W,R,B)** | 83.84% | 100.00% | 96.75% |
| | **Avg(W,R,B)** | ***72.79%*** | 99.90% | ***90.48%*** |
| | **Diff(Min*,Max*)** | ***27.11%*** | | ***12.91%*** |
| **PWC95 B** | **Walking (W)** | 29.25% | 100.00% | 67.02% |
| | **Running (R)** | 38.54% | 100.00% | 88.77% |
| | **Bicycle (B)** | 56.25% | 99.40% | 87.09% |
| | **Min(W,R,B)** | 29.25% | 99.40% | 67.02% |
| | **Max(W,R,B)** | 56.25% | 100.00% | 88.77% |
| | **Avg(W,R,B)** | ***41.35%*** | 99.80% | ***80.96%*** |
| | **Diff(Min*,Max*)** | ***58.45%*** | | ***21.75%*** |
| **WM50 B** | **Walking (W)** | 21.69% | 100.00% | 73.63% |
| | **Running (R)** | 4.89% | 100.00% | 86.43% |
| | **Bicycle (B)** | 83.44% | 100.00% | 83.44% |
| | **Min(W,R,B)** | 4.89% | 100.00% | 73.63% |
| | **Max(W,R,B)** | 83.44% | 100.00% | 86.43% |
| | **Avg(W,R,B)** | ***36.67%*** | 100.00% | ***81.17%*** |
| | **Diff(Min*,Max*)** | ***63.33%*** | | ***12.80%*** |
| | | | | Continued on next page |

**Table C.3 – continued from previous page**

| Cross Guess | | P | | NP |
|---|---|---|---|---|
| | | min | max | value |
| PWM60 | Walking (W) | 95.22% | 100.00% | 96.16% |
| | Running (R) | 64.69% | 95.05% | 87.00% |
| | Bicycle (B) | 83.19% | 100.00% | 83.19% |
| | Min(W,R,B) | 64.69% | 95.05% | 83.19% |
| | Max(W,R,B) | 95.22% | 100.00% | 96.16% |
| | Avg(W,R,B) | ***81.03%*** | 98.35% | ***88.78%*** |
| | Diff(Min*,Max*) | ***17.32%*** | | ***12.97%*** |
| PWC55 | Walking (W) | 94.30% | 100.00% | 95.84% |
| | Running (R) | 23.37% | 98.40% | 87.20% |
| | Bicycle (B) | 89.00% | 98.96% | 89.00% |
| | Min(W,R,B) | 23.37% | 98.40% | 87.20% |
| | Max(W,R,B) | 94.30% | 100.00% | 95.84% |
| | Avg(W,R,B) | ***68.89%*** | 99.12% | ***90.68%*** |
| | Diff(Min*,Max*) | ***30.23%*** | | ***8.64%*** |
| WM70 | Walking (W) | 92.11% | 100.00% | 94.73% |
| | Running (R) | 34.71% | 94.79% | 85.00% |
| | Bicycle (B) | 89.75% | 96.99% | 89.75% |
| | Min(W,R,B) | 34.71% | 94.79% | 85.00% |
| | Max(W,R,B) | 92.11% | 100.00% | 94.73% |
| | Avg(W,R,B) | ***72.19%*** | 97.26% | ***89.83%*** |
| | Diff(Min*,Max*) | ***25.07%*** | | ***9.73%*** |

**Table C.4:** Classifications on all placements for each method for walking

| | | ALL | FP | BP | ARM | BAG | HAND |
|---|---|---|---|---|---|---|---|
| **PWM75 B** | **M** | 28.35% | 61.50% | 100.00% | 100.00% | 96.00% | 73.50% |
| | **C** | 44.65% | 51.54% | 66.67% | 66.67% | 65.48% | 49.40% |
| | **CG** | 66.97% | 77.31% | 100.00% | 100.00% | 98.22% | 24.10% |
| **PWC5 B** | **M** | 60.47% | 18.28% | 100.00% | 100.00% | 88.89% | 34.94% |
| | **C** | 64.50% | 37.20% | 66.67% | 66.67% | 65.33% | 53.41% |
| | **CG** | 96.75% | 55.81% | 100.00% | 100.00% | 98.00% | 80.12% |
| **PWC95 B** | **M** | 29.17% | 32.04% | 100.00% | 100.00% | 96.89% | 83.13% |
| | **C** | 44.68% | 44.73% | 66.67% | 63.19% | 65.78% | 54.62% |
| | **CG** | 67.02% | 29.25% | 100.00% | 94.79% | 98.67% | 81.93% |
| **WM50 B** | **M** | 34.76% | 48.00% | 100.00% | 100.00% | 91.11% | 87.95% |
| | **C** | 49.09% | 43.66% | 66.67% | 66.67% | 65.48% | 47.79% |
| | **CG** | 73.63% | 65.48% | 100.00% | 100.00% | 98.22% | 21.69% |
| **PWM60** | **M** | 79.01% | 74.36% | 98.43% | 96.43% | 91.06% | 96.49% |
| | **C** | 64.11% | 63.48% | 63.60% | 66.67% | 63.86% | 65.23% |
| | **CG** | 96.16% | 95.22% | 95.40% | 100.00% | 95.79% | 97.84% |
| **PWC55** | **M** | 81.43% | 73.10% | 99.22% | 96.43% | 94.21% | 97.57% |
| | **C** | 63.90% | 63.46% | 62.93% | 66.67% | 64.40% | 64.59% |
| | **CG** | 95.84% | 95.19% | 94.30% | 100.00% | 96.60% | 96.89% |
| **WM70** | **M** | 82.36% | 80.82% | 98.65% | 100.00% | 94.21% | 87.57% |
| | **C** | 63.15% | 61.40% | 65.69% | 66.67% | 64.74% | 63.87% |
| | **CG** | 94.73% | 92.11% | 98.54% | 100.00% | 97.11% | 95.81% |

**Table C.5:** Classifications on all placements for each method for running

| | | ALL | FP | BP | ARM | BAG | HAND |
|---|---|---|---|---|---|---|---|
| PWM75 B | M | 66.97% | 56.56% | 80.12% | 100.00% | 15.11% | 87.95% |
| | C | 56.98% | 57.28% | 63.35% | 60.42% | 42.96% | 66.67% |
| | CG | 85.47% | 85.91% | 95.03% | 90.63% | 64.44% | 100.00% |
| PWC5 B | M | 89.33% | 74.41% | 73.91% | 97.92% | 48.44% | 85.54% |
| | C | 60.57% | 52.47% | 60.87% | 66.67% | 64.15% | 65.86% |
| | CG | 90.85% | 78.71% | 91.30% | 100.00% | 96.22% | 98.80% |
| PWC95 B | M | 64.84% | 56.77% | 89.44% | 93.75% | 28.89% | 93.98% |
| | C | 59.18% | 51.90% | 64.39% | 25.69% | 40.30% | 66.67% |
| | CG | 88.77% | 77.85% | 96.58% | 38.54% | 60.44% | 100.00% |
| WM50 B | M | 68.70% | 55.48% | 75.78% | 95.83% | 4.00% | 100.00% |
| | C | 57.62% | 40.22% | 56.94% | 65.28% | 9.19% | 66.67% |
| | CG | 86.43% | 60.32% | 85.40% | 97.92% | 4.89% | 100.00% |
| PWM60 | M | 59.60% | 82.88% | 64.69% | 75.08% | 0.89% | 86.36% |
| | C | 58.00% | 62.72% | 43.13% | 57.59% | 57.63% | 63.37% |
| | CG | 87.00% | 94.08% | 64.69% | 86.38% | 86.44% | 95.05% |
| PWC55 | M | 61.87% | 92.15% | 56.61% | 78.41% | 4.44% | 93.85% |
| | C | 58.13% | 63.23% | 36.23% | 56.42% | 48.74% | 65.60% |
| | CG | 87.20% | 94.84% | 23.37% | 84.63% | 73.11% | 98.40% |
| WM70 | M | 50.55% | 92.03% | 30.57% | 84.22% | 0.44% | 86.36% |
| | C | 56.65% | 61.86% | 33.53% | 57.42% | 36.74% | 63.19% |
| | CG | 85.00% | 92.79% | 34.71% | 86.13% | 55.11% | 94.79% |

**Table C.6:** Classifications on all placements for each method for bike

| | | ALL | FP | BP | ARM | BAG | HAND |
|---|---|---|---|---|---|---|---|
| PWM75 B | M | 71.65% | 91.83% | 91.93% | 95.83% | 92.89% | 100.00% |
| | C | 57.55% | 63.08% | 65.01% | 64.58% | 62.37% | 66.27% |
| | CG | 86.33% | 94.62% | 97.52% | 96.88% | 93.56% | 99.40% |
| PWC5 B | M | 58.44% | 46.45% | 85.09% | 87.50% | 92.89% | 91.57% |
| | C | 55.89% | 63.15% | 66.46% | 65.97% | 60.44% | 66.27% |
| | CG | 83.84% | 94.73% | 99.69% | 98.96% | 90.67% | 99.40% |
| PWC95 B | M | 68.90% | 92.47% | 95.03% | 95.83% | 91.56% | 98.80% |
| | C | 58.06% | 64.51% | 64.60% | 37.50% | 65.78% | 66.27% |
| | CG | 87.09% | 96.77% | 96.89% | 56.25% | 98.67% | 99.40% |
| WM50 B | M | 64.33% | 91.83% | 93.17% | 97.92% | 95.11% | 98.80% |
| | C | 55.62% | 64.09% | 64.18% | 63.89% | 62.67% | 66.67% |
| | CG | 83.44% | 96.13% | 96.27% | 95.83% | 94.00% | 100.00% |
| PWM60 | M | 42.13% | 87.96% | 77.64% | 100.00% | 87.40% | 69.88% |
| | C | 55.46% | 61.94% | 61.28% | 66.67% | 63.73% | 65.06% |
| | CG | 83.19% | 92.90% | 91.93% | 100.00% | 95.60% | 97.59% |
| PWC55 | M | 44.06% | 89.03% | 79.50% | 100.00% | 89.30% | 69.88% |
| | C | 59.30% | 61.29% | 60.87% | 65.97% | 64.39% | 65.06% |
| | CG | 89.00% | 91.94% | 91.31% | 98.96% | 96.58% | 97.59% |
| WM70 | M | 59.87% | 88.39% | 92.55% | 89.58% | 91.31% | 75.90% |
| | C | 59.83% | 61.43% | 61.08% | 63.19% | 64.55% | 64.66% |
| | CG | 89.75% | 92.15% | 91.61% | 94.79% | 96.82% | 96.99% |

# Appendix D

# Features

**Table D.1:** Table showing all features and their associated number

| Number | Feature |
|---|---|
| 1 | $\text{mean}(\hat{X})$ |
| 2 | $\text{mean}(\hat{Y})$ |
| 3 | $\text{mean}(\hat{Z})$ |
| 4 | mean(*MAG*) |
| 5 | $\text{mean}(\text{FFT}(\hat{X}))$ |
| 6 | $\text{mean}(\text{FFT}(\hat{Y}))$ |
| 7 | $\text{mean}(\text{FFT}(\hat{Z}))$ |
| 8 | mean(FFT(*MAG*)) |
| 9 | $\text{var}(\hat{X})$ |
| 10 | $\text{var}(\hat{Y})$ |
| 11 | $\text{var}(\hat{Z})$ |
| 12 | var(*MAG*) |
| 13 | $\text{var}(\text{FFT}(\hat{X}))$ |
| | Continued on next page |

**Table D.1 – continued from previous page**

| Number | Feature |
|---|---|
| 14 | $\mathrm{var}(\mathrm{FFT}(\hat{Y}))$ |
| 15 | $\mathrm{var}(\mathrm{FFT}(\hat{Z}))$ |
| 16 | $\mathrm{var}(\mathrm{FFT}(\boldsymbol{MAG}))$ |
| 17 | $\mathrm{var}(\mathrm{real}(\mathrm{FFT}(\hat{X})))$ |
| 18 | $\mathrm{var}(\mathrm{real}(\mathrm{FFT}(\hat{Y})))$ |
| 19 | $\mathrm{var}(\mathrm{real}(\mathrm{FFT}(\hat{Z})))$ |
| 20 | $\mathrm{var}(\mathrm{real}(\mathrm{FFT}(\boldsymbol{MAG})))$ |
| 21 | $\mathrm{var}(\mathrm{abs}(\mathrm{FFT}(\hat{X})))$ |
| 22 | $\mathrm{var}(\mathrm{abs}(\mathrm{FFT}(\hat{Y})))$ |
| 23 | $\mathrm{var}(\mathrm{abs}(\mathrm{FFT}(\hat{Z})))$ |
| 24 | $\mathrm{var}(\mathrm{abs}(\mathrm{FFT}(\boldsymbol{MAG})))$ |
| 25 | $\mathrm{var}(\mathrm{FFT}(\hat{X})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{X})))$ |
| 26 | $\mathrm{var}(\mathrm{FFT}(\hat{Y})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{Y})))$ |
| 27 | $\mathrm{var}(\mathrm{FFT}(\hat{Z})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{Z})))$ |
| 28 | $\mathrm{var}(\mathrm{FFT}(\boldsymbol{MAG})\cdot\mathrm{conj}(\mathrm{FFT}(\boldsymbol{MAG})))$ |
| 29 | $\mathrm{var}(\mathrm{DCT}(\hat{X})))$ |
| 30 | $\mathrm{var}(\mathrm{DCT}(\hat{Y})))$ |
| 31 | $\mathrm{var}(\mathrm{DCT}(\hat{Z})))$ |
| 32 | $\mathrm{var}(\mathrm{DCT}(\boldsymbol{MAG})))$ |
| 33 | $\mathrm{E}(1,1){:}[\mathrm{E},\mathrm{V}] = (\mathrm{eig}(\mathrm{cov}(\boldsymbol{A})))$ |
| 34 | $\mathrm{E}(2,2){:}[\mathrm{E},\mathrm{V}] = (\mathrm{eig}(\mathrm{cov}(\boldsymbol{A})))$ |
| 35 | $\mathrm{E}(3,3){:}[\mathrm{E},\mathrm{V}] = (\mathrm{eig}(\mathrm{cov}(\boldsymbol{A})))$ |
| 36 | $\mathrm{E}(1,1){:}[\mathrm{E},\mathrm{V}] = (\mathrm{eig}(\mathrm{cov}((\mathrm{FFT}(\boldsymbol{A})))))$ |
| 37 | $\mathrm{E}(2,2){:}[\mathrm{E},\mathrm{V}] = (\mathrm{eig}(\mathrm{cov}((\mathrm{FFT}(\boldsymbol{A})))))$ |

Continued on next page

**Table D.1 – continued from previous page**

| Number | Feature |
|---|---|
| 38 | $\mathrm{E}\,(3,3){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}((\mathrm{FFT}(A)))))$ |
| 39 | $\mathrm{E}\,(1,1){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}(\mathrm{real}\,(\mathrm{FFT}(A)))))$ |
| 40 | $\mathrm{E}\,(2,2){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}(\mathrm{real}\,(\mathrm{FFT}(A)))))$ |
| 41 | $\mathrm{E}\,(3,3){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}(\mathrm{real}\,(\mathrm{FFT}(A)))))$ |
| 42 | $\mathrm{E}\,(1,1){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}(\mathrm{abs}(\mathrm{FFT}(A)))))$ |
| 43 | $\mathrm{E}\,(2,2){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}(\mathrm{abs}(\mathrm{FFT}(A)))))$ |
| 44 | $\mathrm{E}\,(3,3){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}(\mathrm{abs}(\mathrm{FFT}(A)))))$ |
| 45 | $\mathrm{E}\,(1,1){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}(\mathrm{FFT}(A)\cdot\mathrm{conj}(\mathrm{FFT}(A)))))$ |
| 46 | $\mathrm{E}\,(2,2){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}(\mathrm{FFT}(A)\cdot\mathrm{conj}(\mathrm{FFT}(A)))))$ |
| 47 | $\mathrm{E}\,(3,3){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}(\mathrm{FFT}(A)\cdot\mathrm{conj}(\mathrm{FFT}(A)))))$ |
| 48 | $\mathrm{E}\,(1,1){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}((\mathrm{DCT}(A)))))$ |
| 49 | $\mathrm{E}\,(2,2){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}((\mathrm{DCT}(A)))))$ |
| 50 | $\mathrm{E}\,(3,3){:}[\mathrm{E,V}] = (\mathrm{eig}(\mathrm{cov}((\mathrm{DCT}(A)))))$ |
| 51 | $\max(\hat{X})$ |
| 52 | $\max(\hat{Y})$ |
| 53 | $\max(\hat{Z})$ |
| 54 | $\max(MAG)$ |
| 55 | $\max(\mathrm{FFT}(\hat{X}))$ |
| 56 | $\max(\mathrm{FFT}(\hat{Y}))$ |
| 57 | $\max(\mathrm{FFT}(\hat{Z}))$ |
| 58 | $\max(\mathrm{FFT}(MAG))$ |
| 59 | $\max(\mathrm{real}(\mathrm{FFT}(\hat{X})))$ |
| 60 | $\max(\mathrm{real}(\mathrm{FFT}(\hat{Y})))$ |
| 61 | $\max(\mathrm{real}(\mathrm{FFT}(\hat{Z})))$ |

Continued on next page

**Table D.1 – continued from previous page**

| Number | Feature |
|---|---|
| 62 | $\max(\text{real}(\text{FFT}(MAG)))$ |
| 63 | $\max(\text{abs}(\text{FFT}(\hat{X})))$ |
| 64 | $\max(\text{abs}(\text{FFT}(\hat{Y})))$ |
| 65 | $\max(\text{abs}(\text{FFT}(\hat{Z})))$ |
| 66 | $\max(\text{abs}(\text{FFT}(MAG)))$ |
| 67 | $\max(\text{FFT}(\hat{X})\cdot\text{conj}(\text{FFT}(\hat{X})))$ |
| 68 | $\max(\text{FFT}(\hat{Y})\cdot\text{conj}(\text{FFT}(\hat{Y})))$ |
| 69 | $\max(\text{FFT}(\hat{Z})\cdot\text{conj}(\text{FFT}(\hat{Z})))$ |
| 70 | $\max(\text{FFT}(MAG)\cdot\text{conj}(\text{FFT}(MAG)))$ |
| 71 | $\max(\text{DCT}(\hat{X}))$ |
| 72 | $\max(\text{DCT}(\hat{Y}))$ |
| 73 | $\max(\text{DCT}(\hat{Z}))$ |
| 74 | $\max(\text{DCT}(MAG))$ |
| 75 | energy($\hat{X}$) |
| 76 | energy($\hat{Y}$) |
| 77 | energy($\hat{Z}$) |
| 78 | energy(*MAG*) |
| 79 | energy($\text{FFT}(\hat{X})$) |
| 80 | energy($\text{FFT}(\hat{Y})$) |
| 81 | energy($\text{FFT}(\hat{Z})$) |
| 82 | energy($\text{FFT}(MAG)$) |
| 83 | energy($\text{real}(\text{FFT}(\hat{X}))$) |
| 84 | energy($\text{real}(\text{FFT}(\hat{Y}))$) |
| 85 | energy($\text{real}(\text{FFT}(\hat{Z}))$) |

Continued on next page

**Table D.1 – continued from previous page**

| Number | Feature |
|---|---|
| 86 | energy($\mathrm{real}\,(\mathrm{FFT}(MAG))$) |
| 87 | energy($\mathrm{abs}(\mathrm{FFT}(\hat{X}))$) |
| 88 | energy($\mathrm{abs}(\mathrm{FFT}(\hat{Y}))$) |
| 89 | energy($\mathrm{abs}(\mathrm{FFT}(\hat{Z}))$) |
| 90 | energy($\mathrm{abs}(\mathrm{FFT}(MAG))$) |
| 91 | energy($\mathrm{FFT}(\hat{X})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{X}))$) |
| 92 | energy($\mathrm{FFT}(\hat{Y})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{Y}))$) |
| 93 | energy($\mathrm{FFT}(\hat{Z})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{Z}))$) |
| 94 | energy($\mathrm{FFT}(MAG)\cdot\mathrm{conj}(\mathrm{FFT}(MAG))$) |
| 95 | energy($\mathrm{DCT}(\hat{X})$) |
| 96 | energy($\mathrm{DCT}(\hat{Y})$) |
| 97 | energy($\mathrm{DCT}(\hat{Z})$) |
| 98 | energy($\mathrm{DCT}(MAG)$) |
| 99 | entropy($\hat{X}$) |
| 100 | entropy($\hat{Y}$) |
| 101 | entropy($\hat{Z}$) |
| 102 | entropy(*MAG*) |
| 103 | entropy($\mathrm{FFT}(\hat{X})$) |
| 104 | entropy($\mathrm{FFT}(\hat{Y})$) |
| 105 | entropy($\mathrm{FFT}(\hat{Z})$) |
| 106 | entropy($\mathrm{FFT}(MAG)$) |
| 107 | entropy($\mathrm{real}\,(\mathrm{FFT}(\hat{X}))$) |
| 108 | entropy($\mathrm{real}\,(\mathrm{FFT}(\hat{Y}))$) |
| 109 | entropy($\mathrm{real}\,(\mathrm{FFT}(\hat{Z}))$) |

Continued on next page

**Table D.1 – continued from previous page**

| Number | Feature |
|---|---|
| 110 | entropy(real(FFT(*MAG*))) |
| 111 | entropy(abs(FFT($\hat{X}$))) |
| 112 | entropy(abs(FFT($\hat{Y}$))) |
| 113 | entropy(abs(FFT($\hat{Z}$))) |
| 114 | entropy(abs(FFT(*MAG*))) |
| 115 | entropy(FFT($\hat{X}$)·conj(FFT($\hat{X}$))) |
| 116 | entropy(FFT($\hat{Y}$)·conj(FFT($\hat{Y}$))) |
| 117 | entropy(FFT($\hat{Z}$)·conj(FFT($\hat{Z}$))) |
| 118 | entropy(FFT(*MAG*)·conj(FFT(*MAG*))) |
| 119 | entropy(DCT($\hat{X}$)) |
| 120 | entropy(DCT($\hat{Y}$)) |
| 121 | entropy(DCT($\hat{Z}$)) |
| 122 | entropy(DCT(*MAG*)) |
| 123 | mean(MRV(1:3)($\hat{X}$)) |
| 124 | mean(MRV(1:3)($\hat{Y}$)) |
| 125 | mean(MRV(1:3)($\hat{Z}$)) |
| 126 | mean(MRV(1:3)(*MAG*)) |
| 127 | mean(MRV(1:3)(FFT($\hat{X}$))) |
| 128 | mean(MRV(1:3)(FFT($\hat{Y}$))) |
| 129 | mean(MRV(1:3)(FFT($\hat{Z}$))) |
| 130 | mean(MRV(1:3)(FFT(*MAG*))) |
| 131 | mean(MRV(1:3)(real(FFT($\hat{X}$)))) |
| 132 | mean(MRV(1:3)(real(FFT($\hat{Y}$)))) |
| 133 | mean(MRV(1:3)(real(FFT($\hat{Z}$)))) |

Continued on next page

Table D.1 – continued from previous page

| Number | Feature |
|---|---|
| 134 | mean(MRV(1:3)(real(FFT(*MAG*)))) |
| 135 | mean(MRV(1:3)(abs(FFT($\hat{X}$)))) |
| 136 | mean(MRV(1:3)(abs(FFT($\hat{Y}$)))) |
| 137 | mean(MRV(1:3)(abs(FFT($\hat{Z}$)))) |
| 138 | mean(MRV(1:3)(abs(FFT(*MAG*)))) |
| 139 | mean(MRV(1:3)(FFT($\hat{X}$)·conj(FFT($\hat{X}$)))) |
| 140 | mean(MRV(1:3)(FFT($\hat{Y}$)·conj(FFT($\hat{Y}$)))) |
| 141 | mean(MRV(1:3)(FFT($\hat{Z}$)·conj(FFT($\hat{Z}$)))) |
| 142 | mean(MRV(1:3)(FFT(*MAG*)·conj(FFT(*MAG*)))) |
| 143 | mean(MRV(1:3)(DCT($\hat{X}$))) |
| 144 | mean(MRV(1:3)(DCT($\hat{Y}$))) |
| 145 | mean(MRV(1:3)(DCT($\hat{Z}$))) |
| 146 | mean(MRV(1:3)(DCT(*MAG*))) |
| 147 | ZCR(FFT($\hat{X}$)) |
| 148 | ZCR(FFT($\hat{Y}$)) |
| 149 | ZCR(FFT($\hat{Z}$)) |
| 150 | ZCR(FFT(*MAG*)) |
| 151 | ZCR(FFT($\hat{X}$)) |
| 152 | ZCR(DCT($\hat{Y}$)) |
| 153 | ZCR(DCT($\hat{Z}$)) |
| 154 | ZCR(DCT(*MAG*)) |
| 155 | THD($\hat{X}$) |
| 156 | THD($\hat{Y}$) |
| 157 | THD($\hat{Z}$) |

Continued on next page

**Table D.1 – continued from previous page**

| Number | Feature |
|---|---|
| 158 | THD(*MAG*) |
| 159 | $\mathrm{THD}(\mathrm{real}(\mathrm{FFT}(\hat{X})))$ |
| 160 | $\mathrm{THD}(\mathrm{real}(\mathrm{FFT}(\hat{Y})))$ |
| 161 | $\mathrm{THD}(\mathrm{real}(\mathrm{FFT}(\hat{Z}))))$ |
| 162 | THD(real(FFT(*MAG*))) |
| 163 | $\mathrm{THD}(\mathrm{abs}(\mathrm{FFT}(\hat{X})))$ |
| 164 | $\mathrm{THD}(\mathrm{abs}(\mathrm{FFT}(\hat{Y})))$ |
| 165 | $\mathrm{THD}(\mathrm{abs}(\mathrm{FFT}(\hat{Z})))$ |
| 166 | THD(abs(FFT(*MAG*))) |
| 167 | $\mathrm{THD}(\mathrm{FFT}(\hat{X})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{X})))$ |
| 168 | $\mathrm{THD}(\mathrm{FFT}(\hat{Y})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{Y})))$ |
| 169 | $\mathrm{THD}(\mathrm{FFT}(\hat{Z})\cdot\mathrm{conj}(\mathrm{FFT}(\hat{Z})))$ |
| 170 | THD(FFT(*MAG*)·conj(FFT(*MAG*))) |

# Bibliography


[1] Reza Akhavian and Amir H. Behzadan. "Construction Activity Recognition for Simulation Input Modeling Using Machine Learning Classifiers". In: *Proceedings of the 2014 Winter Simulation Conference*. WSC '14. Savannah, Georgia: IEEE Press, 2014, pp. 3296–3307. URL: http://dl.acm.org/citation.cfm?id=2693848.2694261.

[2] Ling Bao and Stephen S. Intille. "Activity Recognition from User-Annotated Acceleration Data." In: *Pervasive*. Ed. by Alois Ferscha and Friedemann Mattern. Vol. 3001. Lecture Notes in Computer Science. Springer, 2004, pp. 1–17. ISBN: 3-540-21835-1. URL: http://dblp.uni-trier.de/db/conf/pervasive/pervasive2004.html#BaoI04.

[3] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data". In: *SIGKDD Explor. Newsl.* 6.1 (June 2004), pp. 20–29. ISSN: 1931-0145. DOI: 10.1145/1007730.1007735. URL: http://doi.acm.org/10.1145/1007730.1007735.

[4] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[5] C. M. Bishop. *Pattern recognition and machine learning*. New York: Springer, 2006.

[6] J. Farringdon et al. "Wearable sensor badge and sensor jacket for context awareness". In: *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*. 1999, pp. 107–113. DOI: 10.1109/ISWC.1999.806681.

[7] Davide Figo et al. "Preprocessing Techniques for Context Recognition from Accelerometer Data". In: *Personal Ubiquitous Comput.* 14.7 (2010), pp. 645–662. ISSN: 1617-4909. DOI: 10.1007/s00779-010-0293-9. URL: http://dx.doi.org/10.1007/s00779-010-0293-9.

[8] Giles M. Foody and Ajay Mathur. “Toward intelligent training of supervised image classifications: directing training data acquisition for {SVM} classification”. In: *Remote Sensing of Environment* 93.1–2 (2004), pp. 107 –117. ISSN: 0034-4257. DOI: http://dx.doi.org/10.1016/j.rse.2004.06.017. URL: http://www.sciencedirect.com/science/article/pii/S003442570400207X.

[9] Giles M. Foody et al. “Training set size requirements for the classification of a specific class”. In: *Remote Sensing of Environment* 104.1 (2006), pp. 1 –14. ISSN: 0034-4257. DOI: http://dx.doi.org/10.1016/j.rse.2006.03.004. URL: http://www.sciencedirect.com/science/article/pii/S0034425706001234.

[10] StephenS. Intille et al. “Tools for Studying Behavior and Technology in Natural Settings”. English. In: *UbiComp 2003: Ubiquitous Computing*. Ed. by AnindK. Dey, Albrecht Schmidt, and JosephF. McCarthy. Vol. 2864. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 157–174. ISBN: 978-3-540-20301-8. DOI: 10.1007/978-3-540-39653-6_13. URL: http://dx.doi.org/10.1007/978-3-540-39653-6_13.

[11] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. “Activity Recognition Using Cell Phone Accelerometers”. In: *SIGKDD Explor. Newsl.* 12.2 (2011), pp. 74–82. ISSN: 1931-0145. DOI: 10.1145/1964897.1964918. URL: http://doi.acm.org/10.1145/1964897.1964918.

[12] Seon-Woo Lee and Kenji Mase. “Activity and location recognition using wearable sensors”. In: *IEEE pervasive computing* 1.3 (2002), pp. 24–32.

[13] Jesper Lekland and Asmir Sabanovic. *User Independent Gym Exercise Recognition Using a Single Wrist Worn Accelerometer*. 2014.

[14] Yijuan Lu et al. “Feature Selection Using Principal Feature Analysis”. In: *Proceedings of the 15th International Conference on Multimedia*. MULTIMEDIA ’07. Augsburg, Germany: ACM, 2007, pp. 301–304. ISBN: 978-1-59593-702-5. DOI: 10.1145/1291233.1291297. URL: http://doi.acm.org/10.1145/1291233.1291297.

[15] MathWorks. *Absolute value and complex magnitude*. 2006. URL: http://se.mathworks.com/help/matlab/ref/abs.html.

[16] MathWorks. *Average or mean value of array*. 2006. URL: http://se.mathworks.com/help/matlab/ref/mean.html.

[17] MathWorks. *Complex conjugate*. 2006. URL: http://se.mathworks.com/help/matlab/ref/conj.html.

[18] MathWorks. *Covariance*. 2006. URL: http://se.mathworks.com/help/matlab/ref/cov.html.

[19] MathWorks. *Discrete cosine transform*. 2006. URL: http://se.mathworks.com/help/signal/ref/dct.html.

[20] MathWorks. *Fast Fourier transform*. 2006. URL: http://se.mathworks.com/help/matlab/ref/fft.html.

[21] MathWorks. *Histogram Plot*. 2006.

[22] MathWorks. *Histogram plot*. 2014. URL: http://se.mathworks.com/help/matlab/ref/histogram.html.

[23] MathWorks. *Largest elements in array*. 2006. URL: http://se.mathworks.com/help/matlab/ref/max.html.

[24] MathWorks. *Principal component analysis of raw data*. 2012. URL: http://se.mathworks.com/help/stats/pca.html.

[25] MathWorks. *Real part of complex number*. 2006. URL: http://se.mathworks.com/help/matlab/ref/real.html.

[26] MathWorks. *Replace Discouraged Instances of hist and histc*. 2014. URL: http://se.mathworks.com/help/matlab/creating_plots/replace-discouraged-instances-of-hist-and-histc.html.

[27] MathWorks. *Total harmonic distortion*. 2013. URL: http://se.mathworks.com/help/signal/ref/thd.html.

[28] MathWorks. *Variance*. 2006. URL: http://se.mathworks.com/help/matlab/ref/var.html.

[29] Angshu Rai et al. "Mining Complex Activities in the Wild via a Single Smartphone Accelerometer". In: *Proceedings of the Sixth International Workshop on Knowledge Discovery from Sensor Data*. SensorKDD '12. Beijing, China: ACM, 2012, pp. 43–51. ISBN: 978-1-4503-1554-8. DOI: 10.1145/2350182.2350187. URL: http://doi.acm.org/10.1145/2350182.2350187.

[30] Nishkam Ravi et al. "Activity recognition from accelerometer data". In: *IAAI'05: Proceedings of the 17th conference on Innovative applications of artificial intelligence*. Pittsburgh, Pennsylvania: AAAI Press, 2005, pp. 1541–1546. ISBN: 1-57735-236-x.

[31] Mandeep Singh and Amandeep Cheema. "Article: Heart Sounds Classification using Feature Extraction of Phonocardiography Signal". In: *International Journal of Computer Applications* 77.4 (2013), pp. 13–17.

[32] Fengxi Song, Zhongwei Guo, and Dayong Mei. "Feature Selection Using Principal Component Analysis". In: *Proceedings of the 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization - Volume 01*. ICSEM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 27–30. ISBN: 978-0-7695-4223-2. DOI: 10.1109/ICSEM.2010.14. URL: http://dx.doi.org/10.1109/ICSEM.2010.14.

[33] Statistics How To. *Choose Bin Sizes for Histograms in Easy Steps*. 2015. URL: http://www.statisticshowto.com/choose-bin-sizes-statistics/.

[34] Gary M. Weiss and Jeffrey W. Lockhart. "Identifying User Traits by Mining Smart Phone Accelerometer Data". In: *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data*. SensorKDD '11. San Diego, California: ACM, 2011, pp. 61–69. ISBN: 978-1-4503-0832-8. DOI: 10.1145/2003653.2003660. URL: http://doi.acm.org/10.1145/2003653.2003660.

[35] Wikipedia. *Confusion matrix — Wikipedia, The Free Encyclopedia*. [Online; accessed 23-October-2015]. 2015. URL: https://en.wikipedia.org/w/index.php?title=Confusion_matrix&oldid=679360886.

[36] Wikipedia. *Data collection — Wikipedia, The Free Encyclopedia*. [Online; accessed 20-October-2015]. 2015. URL: https://en.wikipedia.org/w/index.php?title=Data_collection&oldid=684534858.

[37] Wikipedia. *Dimensionality reduction — Wikipedia, The Free Encyclopedia*. [Online; accessed 7-October-2015]. 2015. URL: https://en.wikipedia.org/w/index.php?title=Dimensionality_reduction&oldid=678334941.

[38] Wikipedia. *Feature (machine learning) — Wikipedia, The Free Encyclopedia*. [Online; accessed 13-October-2015]. 2015. URL: https://en.wikipedia.org/w/index.php?title=Feature_(machine_learning)&oldid=675707570.

[39] Wikipedia. *Feature selection — Wikipedia, The Free Encyclopedia*. [Online; accessed 28-September-2015]. 2015. URL: https://en.wikipedia.org/w/index.php?title=Feature_selection&oldid=680919968.

[40] Wikipedia. *Statistical classification — Wikipedia, The Free Encyclopedia*. [Online; accessed 20-October-2015]. 2015. URL: https://en.wikipedia.org/w/index.php?title=Statistical_classification&oldid=681093717.

[41] Wikipedia. *Total harmonic distortion — Wikipedia, The Free Encyclopedia*. [Online; accessed 7-October-2015]. 2015. URL: https://en.wikipedia.org/w/index.php?title=Total_harmonic_distortion&oldid=650239592.

[42] Wikipedia. *Zero-crossing rate — Wikipedia, The Free Encyclopedia*. [Online; accessed 7-October-2015]. 2015. URL: https://en.wikipedia.org/w/index.php?title=Zero-crossing_rate&oldid=672692298.

[43] Jose R Zapata. *Zero Crossing Rate*. 2011. URL: http://www.mathworks.com/matlabcentral/fileexchange/31663-zero-crossing-rate/content/ZCR.m.

Master's Theses in Mathematical Sciences 2015:E50
ISSN 1404-6342

LUTFMA-3284-2015

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/