Title: Sampling rate comparison in accelerometer based HAR

Student: Daniel Castelló Garcia

Problem description:

In the last twenty years, mobile phones have exponentially grown in power and capabilities. Since smartphones tend to have accelerometers inside them, the amount of data on human activity has multiplied. As a consequence, the number of papers on Human Activity Recognition has increased and the areas of research being explored have widened. Among those, papers are found proposing machine-learning solutions with energy-efficient approaches. In Human Activity Recognition studies, two driving causes of battery consumption exist: The accelerometer being used, and the network being used. In order to reduce the energy consumption of the accelerometer, it must be used less, thus reducing the amount of data generated. In order to improve the energy efficiency of the network, compression techniques, such as transmitting the differences in measurements, can be implemented. A reduction in the amount of data gathered by the accelerometer directly reduces the amount of data transmitted. Since this improves both the energy efficiency of the network and of the accelerometer, it is the focus of this master's thesis. The tackled problem is that of setting a standard for the sampling rate of accelerometers used in human activity recognition.

Responsible professor: Frank Alexander Kraemer

Supervisor: Frank Alexander Kraemer, Abdulmajid Murad

Abstract

Human activity recognition aims to identify patterns in data generated through human activity. This activity commonly describes movement and can be gathered through a plethora of sensors. Given their low price and accessibility, accelerometers are frequently the sensor of choice in studies aiming to analyze and classify human activity. When body-worn, these sensors are part of small systems that must gather data and transmit it in real-time through wireless networks, and whose battery usage is of critical importance.

Proposed in this master's thesis is a comparison of systems that aim to recognize and classify human activity, but make use of lowered amounts of data. Fewer data samples are used in order to save battery and allow for lengthier usage of the sensors, but have a negative impact on the performance of the classifier.

By comparing multiple implementations with different parameters, this master's thesis proposes six systems that have near-state-of-the-art performance, with F_1 scores over 87%, and use as little as 1,910 samples to label over 24h of human activity. Compared to previous studies, a loss of 2% precision is accompanied by 30 times more efficient battery usage and is considered a beneficial compromise for future systems.

Throughout this master's thesis over 16,000 different systems are tested, in which different sampling rates, window sizes, and window distances are varied to observe the effects on the system's performance. Certain implementations appear to have common traits that make them more resilient to using lower amounts of samples, such as the usage of longer windows with lengthier window distances.

Acknowledgements

I would like to acknowledge all the opportunities that have been given to me in the past two years and thank NTNU and the IIK department for giving me the chance to start and complete this master's thesis. Frank and Abdulmajid have been key roles in the shaping and guidance of this project, and I would like to thank them for the path they set for me and the freedom I have been given to traverse it. Without any of the mentioned parties this project would not be what it is, and is thanks to them that it is now completed.

I must also thank Lisa, Adolfo, and Maite for their unconditional support and selfless help.

Contents

List of Figures vii

List of Tables xi

List of Acronyms xv

1 Introduction 1
- 1.1 Problem . . . 1
- 1.2 Objective . . . 2
  - 1.2.1 Research questions . . . 2

2 Theoretical knowledge 3
- 2.1 Human activity recognition . . . 3
  - 2.1.1 Data collection . . . 4
  - 2.1.2 Data Pre-processing . . . 5
  - 2.1.3 Data segmentation . . . 5
  - 2.1.4 Feature Generation . . . 6
  - 2.1.5 Classification . . . 9
- 2.2 Machine learning . . . 10
  - 2.2.1 Random Forest . . . 11
- 2.3 Quality metrics . . . 13

3 Background and related work 15
- 3.1 Motivation . . . 15
- 3.2 Literature analysis . . . 15
  - 3.2.1 Papers focused on Human Activity Recognition . . . 16
  - 3.2.2 Papers focused on Adaptive Sampling techniques . . . 18

4 Methodology 21
- 4.1 Implemented system . . . 21
  - 4.1.1 Data Acquisition . . . 22
  - 4.1.2 Pre-processing . . . 23
  - 4.1.3 Segmentation . . . 26

4.1.4 Feature Extraction 27
4.1.5 Classification 27
4.2 Experiments methodology 29
4.2.1 Objectives 29
4.2.2 Experiments 30

5 Experiments **39**
5.1 Experiment 1.1 39
5.2 Experiment 1.2 41
5.2.1 Extension of experiment 1.2: 44
5.3 Experiment 1.3 46
5.3.1 Extension of experiment 1.3: 49
5.4 Experiment 1.4 52
5.5 Experiment 2.1 56
5.6 Experiment 2.2 58
5.7 Experiment 2.3 61
5.8 Experiment 2.4 63

6 Discussion **67**
6.1 Windowless experiments 67
6.2 Windowed experiments 70
6.3 Outcome 72

7 Conclusion **75**
7.1 Contribution 75
7.2 Future work 76

References **79**

List of Figures

2.1 The Activity Recognition Chain, adapted from Bulling et al. [BBS14] . 4
2.2 Fourier analysis of a period square wave. Each row adds a new periodic function. The second column superimposes them; the third column adds them; and the last column shows the amplitude of each periodic function. *Source: https://commons.wikimedia.org/wiki/File:Fourier_synthesis.svg* 9
2.3 The decision tree resulting from training with the data in Table 2.3. . . 12

4.1 The red line denotes the Pareto front, where any point is the optimal value of one quantity in relation to the other. *Source: https://en.wikipedia.org/wiki/File:Pareto_Efficient_Frontier_1024x1024.png* 22
4.2 Representation of the location of the sensors. The red square marks the position of the upper back accelerometer, and the blue square marks the position of the right thigh accelerometer. *Source: https://www.pinterest.com/pin/489907265694622075/* 23
4.3 Confusion matrix of a test showing no instances of either *running*, *lying*, nor *cycling*. 28
4.4 Test with a Forest made of 10 decision trees. F_1 Score == 78.29% . . . 28
4.5 Test with a Forest made of 30 decision trees. F_1 Score == 79.58% . . . 29
4.6 Test with a Forest made of 50 decision trees. F_1 Score == 79.22% . . . 29

5.1 Experiment 1.1: Confusion matrix of the $100Hz$ system. *No. trees = 32; Train samples = 667,000; Test samples = 330,000* 40
5.2 Experiment 1.1: Confusion matrix of the $10Hz$ system. *No. trees = 32; Train samples = 667,000; Test samples = 330,000* 40
5.3 Experiment 1.1: Quality metrics of the $100Hz$ system. *No. trees = 32; Train samples = 667,000; Test samples = 330,000* 40
5.4 Experiment 1.1: Quality metrics of the $10Hz$ system. *No. trees = 32; Train samples = 667,000; Test samples = 330,000* 40
5.5 Experiment 1.1: Quality metrics comparison of experiment 1.1. Even with different sampling rates, all systems used the same amount of total (training and testing) samples. *No. trees = 32; Train samples = 667,000; Test samples = 330,000* 41

5.6 Experiment 1.2: Confusion matrix of the $100Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 333,333* 43
5.7 Experiment 1.2: Confusion matrix of the $1Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 3,333* 43
5.8 Experiment 1.2: Quality metrics of the $100Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 333,333* 43
5.9 Experiment 1.2: Quality metrics of the $1Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 3,333* 43
5.10 Experiment 1.2: Quality metrics comparison of experiment 1.2. Each implementation used the same training set and a reduced test set. *No. trees = 32; Train samples = 666,666; Test samples = (333,333 - 3,333)* 44
5.11 Experiment 1.2 extended: Confusion matrix of the $0.1Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 333* 45
5.12 Experiment 1.2 extended: Confusion matrix of the $0.2Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 666* 45
5.13 Experiment 1.2 extended: Quality metrics of the $0.1Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 333* 45
5.14 Experiment 1.2 extended: Quality metrics of the $0.2Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 666* 45
5.15 Experiment 1.2 extended: Quality metrics comparison of the extension of experiment 1.2. Each implementation used the same training set and a reduced test set. *No. trees = 32; Train samples = 666,666; Test samples = (333 - 2,999)* 46
5.16 Experiment 1.3: Confusion matrix of the 4,666,666 samples system. *No. trees = 32; Train samples = 4,633,333; Test samples = 33,333* 47
5.17 Experiment 1.3: Confusion matrix of the 4,666,666 samples system. *No. trees = 32; Train samples = 4,199,999; Test samples = 466,667* 47
5.18 Experiment 1.3: Quality metrics of the 4,666,666 samples system. *No. trees = 32; Train samples = 4,633,333; Test samples = 33,333* 48
5.19 Experiment 1.3: Quality metrics of the 4,666,666 samples system. *No. trees = 32; Train samples = 4,199,999; Test samples = 466,667* 48
5.20 Experiment 1.3: Quality metrics comparison of experiment 1.3. Blue, red, and purple represent the 33,333 tests; green orange and magenta represent the 10% tests. *No. trees = 32; Train samples = (6,633,333 - 599,999); Test samples = (666,667 - 33,333)* 48
5.21 Experiment 1.3: Difference between the precision, recall, and F_1 score between the systems using the 33,333 samples and the 10% set size for testing. 49
5.22 Experiment 1.3 extended: Confusion matrix of the 66,666 samples system. *No. trees = 32; Train samples = 33,333; Test samples = 33,333* 50

5.23 Experiment 1.3 extended: Confusion matrix of the 66,666 samples system. *No. trees = 32; Train samples = 59,999; Test samples = 6,667* 50

5.24 Experiment 1.3 extended: Quality metrics of the 66,666 samples system. *No. trees = 32; Train samples = 33,333; Test samples = 33,333* 50

5.25 Experiment 1.3 extended: Quality metrics of the 66,666 samples system. *No. trees = 32; Train samples = 59,999; Test samples = 6,667* 51

5.26 Experiment 1.3 extended: Quality metrics comparison of the extension of experiment 1.3. Blue, red, and purple represent the 33,333 tests; green orange and magenta represent the 10% tests. *No. trees = 32; Train samples = (566,665 - 59,999); Test samples = (66,666 - 6,667)* 51

5.27 Experiment 1.3 extended: Difference between the precision, recall, and F_1 score between the systems using the 33,333 samples and the 10% set size for testing. 52

5.28 Experiment 1.4: Quality metrics comparison of the tests using set between $1Hz$ and $90Hz$. *No. trees = 32; Train samples = (8,100,000 - 90,000); Test samples = (900,000 - 10,000)* 53

5.29 Experiment 1.4: Confusion matrix of the $1Hz$ system. *No. trees = 32; Train samples = 90,000; Test samples = 10,000* 54

5.30 Experiment 1.4: Quality metrics of the $1Hz$ system. *No. trees = 32; Train samples = 90,000; Test samples = 10,000* 54

5.31 Experiment 1.4: Confusion matrix of the $8Hz$ system. *No. trees = 32; Train samples = 720,000; Test samples = 80,000* 54

5.32 Experiment 1.4: Quality metrics of the $8Hz$ system. *No. trees = 32; Train samples = 720,000; Test samples = 80,000* 54

5.33 Experiment 1.4: Quality metrics comparison of the tests using set between $0.1Hz$ and $1Hz$. *No. trees = 32; Train samples = (90,000 - 9,000); Test samples = (10,000 - 1,000)* 55

5.34 Experiment 1.4: Quality metrics comparison of the tests using set between $0.1Hz$ and $90Hz$. Note that the graph is not to scale, since the first ten sampling rates implement a logarithmic scale. *No. trees = 32; Train samples = (8,100,000 - 9,000); Test samples = (900,000 - 1,000)* 55

5.35 Experiment 2.1: Confusion matrix of the $1.5s$ window system. *No. trees = 32; Train samples = 44,776; Test samples = 22,054* 57

5.36 Experiment 2.1: Quality metrics of the $1.5s$ window system. *No. trees = 32; Train samples = 44,776; Test samples = 22,054* 57

5.37 Experiment 2.1: Confusion matrix of the $6s$ window system. *No. trees = 32; Train samples = 10,766; Test samples = 5,304* 57

5.38 Experiment 2.1: Quality metrics of the $6s$ window system. *No. trees = 32; Train samples = 10,766; Test samples = 5,304* 57

5.39 Experiment 2.1: Quality metrics comparison of all systems implemented for experiment 2.1. Systems implemented window sizes between $0.5s$ and $10s$, which included between 50 and 1000 samples respectively. *No. trees = 32; Train samples = (135,541 - 6,282); Test samples = (66,760 - 3,095)* 58
5.40 Experiment 2.2: Quality metrics comparison of the tests using windows captured with sampling rates between $2Hz$ and $100Hz$. *No. trees = 32; Train samples = 45,000; Test samples = 22,500* 59
5.41 Experiment 2.2: Confusion matrix of the $44Hz$ system. *No. trees = 32; Train samples = 44,774; Test samples = 22,054* 60
5.42 Experiment 2.2: Quality metrics of the $44Hz$ system. *No. trees = 32; Train samples = 44,774; Test samples = 22,054* 60
5.43 Experiment 2.2: Confusion matrix of the $19Hz$ system. *No. trees = 32; Train samples = 45,581; Test samples = 22,451* 60
5.44 Experiment 2.2: Quality metrics of the $19Hz$ system. *No. trees = 32; Train samples = 45,581; Test samples = 22,451* 60
5.45 Experiment 2.3: F_1 score comparison of all tests. *No. trees = 32; Amount of samples used for the test = (9,022,050 - 1,870)* 62
5.46 Experiment 2.3: Tendency of the F_1 score according to the amount of samples used for training and testing. *No. trees = 32; Amount of samples used for the test = (9,022,050 - 1,870)* 62
5.47 Experiment 2.3: Precision, recall and F_1 score of the best performer in each group. *No. trees = 32; Amount of samples used for the test = (4,252,095 - 1,910)* 63
5.48 Experiment 2.4: F_1 score comparison of all tests. *No. trees = 32; Amount of samples used for the test = (6,014,610 - 1,870)* 65

6.1 Quality metrics comparison of the top performing systems. 74

List of Tables

2.1 Time-domain features 8
2.2 Frequency-domain features 10
2.3 Weather observations and whether or not golf was played. Adapted from [Qui86]. 12
2.4 This table displays a confusion matrix of a binary classifier. True values are instances where both the predicted and actual class coincide: True Positives (TP) and True Negatives (TN). False values indicate wrong predictions: False Positives (FP) and False Negatives (FN). Positives mean that the prediction is Yes, and Negatives the opposite, No. 13
2.5 This table displays a confusion matrix of a HAR classifier. 14

3.1 Tri-axial accelerometer activity recognition examples 16
3.2 Adaptive sampling examples 19

4.1 All the labels originally identified as well as the amount of instances of each, and the percentage relative to the total amount of labeled samples. The *Commute* and *Transport* labels were not described in Hessen and Tessen [HT16] but included in their dataset. 24
4.2 All the labels as they were used by the project. Displays amount of samples, relative amount, and labels included in each class. 26

5.1 Top ten performers and their specifications from experiment 2.3 64

6.1 Classifiers in the resulting Pareto front. 73

List of Acronyms

AI Artificial Inteligence.

ANN Artificial Neural Network.

ARC Activity Recognition Chain.

AS Adaptive Sampling.

CNN Convolutional Neural Network.

FA Fourier analysis.

FN False Negative.

FP False Positive.

HAR Human activity recognition.

Hz Hertz.

ML Machine Learning.

RF Random Forest.

SD Standard Deviation.

SI International System of Units.

TN True Negative.

TP True Positive.

Chapter 1
Introduction

In the last years, the amount of data recorded, transmitted, and stored has grown exponentially. Smartphones' connectivity and capabilities have allowed the concept of Big Data to consolidate itself. Among these capabilities are accelerometers capable of constant recording and sending of acceleration data. A whole range of studies has appeared recently from this increment in human activity data. Specifically, studies that take huge amounts of accelerometer readings and train an artificial intelligence (henceforth AI) so that the AI can make an informed decision on what was the carrier physically doing at any moment. These applications are commonly known as recognizers that perform Human Activity Recognition (HAR).

1.1 Problem

As smartphones become smaller and more computationally powerful, their battery-life naturally decreases. This affects how much data can be gathered with the aforementioned accelerometers. In HAR studies, there are two leading factors for battery consumption: The use of the accelerometers, and the transmission of data through the network. On the one hand, in order to make the network use more efficient, compression techniques are available. On the other hand, using the accelerometer less, directly translates into less power being used by it, as well as the network being used less often. For example, if we move from sampling 20 times a second (20Hz) to a 10Hz sampling rate, we would be using half the energy on the transmission as well as half the energy on the accelerometer sampling.

By reducing the sampling rate in HAR studies, an effect on the system's precision can be noticed. Multiple studies use different sampling rates in order to get the highest precision possible, but there is no consensus on what is the most energy efficient sampling rate. The problem that is seen is that there is no standard when setting a sampling rate for accelerometers in human activity recognition. This causes studies apparently looking for the same, which is identifying human activity through

accelerometer data, to use completely different sampling rates, ranging between 1Hz and 200Hz with similar reported results.

1.2 Objective

Proposed in this thesis is a study of the effect of varying the sampling rate when performing HAR. The objective is that of **evaluating how does lowering the amount of data gathered per second affect the overall precision of the system**. By researching the *impact of sampling at different frequencies on the same system*, a guideline following the efficiency (performance opposed to the amount of samples used) can be established.

Devices engineered to gather and send accelerometer data are small and wearable, given their purpose. Small size limits the available battery-life of these devices. Making them more efficient in their functions is one of the ways to have lengthier and more thorough studies. Nonetheless, losing precision is never wanted in this scenario and limits the possibilities greatly. The loss of precision may be an acceptable trade-off for some studies, and is why the aim of our research is that of *giving a relationship between lowering the sampling rate and the effect on the performance.* Instead of trying to provide a desired rate of measurement, which would depend on each study carried out, a Pareto optimatility is presented so as to allow future studies to select the minimum sampling rate for their desired precision.

1.2.1 Research questions

The objective of this master's thesis is to explore the resilience of Human Activity Recognitors to lower sampling rates.

- **Goal 1:** Train different Random Forest classifiers with variations of the same dataset using distinct sampling rates.
  - **Research question 1:** What is the minimum amount of training data required to have a dependable Random Forest?
  - **Research question 2:** How does the usage of windowing techniques affect the dependence of Random Forests in high sampling rates?
- **Goal 2:** Establish a Pareto Front comparing the performance of all the tested systems with the amount of samples used for training each one of them.
  - **Research question 3:** Which type of machine learning is more vulnerable to lowering the training set's sampling rate?
  - **Research question 4:** What is the most efficient system to be implemented?

Chapter 2
Theoretical knowledge

This master's thesis implements and compares Human Activity Recognition (HAR) systems. Before explaining the designed experiments and the conclusions drawn from them, a theoretical explanation of HAR systems and everything they involve is required. Therefore, this section aims to make clear any knowledge required to be able to understand the other sections.

2.1 Human activity recognition

In computer science, HAR aims to identify human data segments and label them according to a subset of activities. In *A survey on human activity recognition using wearable sensors* written by Lara and Labrador [LL13] the HAR problem is described as follows:

> Given a set $S = \{S_0, ..., S_{k-1}\}$ of k time series, each one from a particular measured attribute, and all defined within time interval $I = [t\alpha, t\omega]$, the goal is to find a temporal partition $\langle I_0, ..., I_{r-1} \rangle$ of I, based on the data in S, and a set of labels representing the activity performed during each interval I_j (e.g., sitting, walking, etc.). This implies that time intervals I_j are consecutive, non-empty, non-overlapping, and such that $\cup_{j=o}^{r-1} I_j = I$.

Activities are expected to be unique, identifiable, and non-simultaneous; and the HAR task is that of labeling them correctly. In order to tackle this task Bulling et al. [BBS14] define a method called Activity Recognition Chain (henceforth ARC), shown in Figure 2.1. The ARC procedure outlines five subsequent steps that transform a raw data input into a class label of that input.

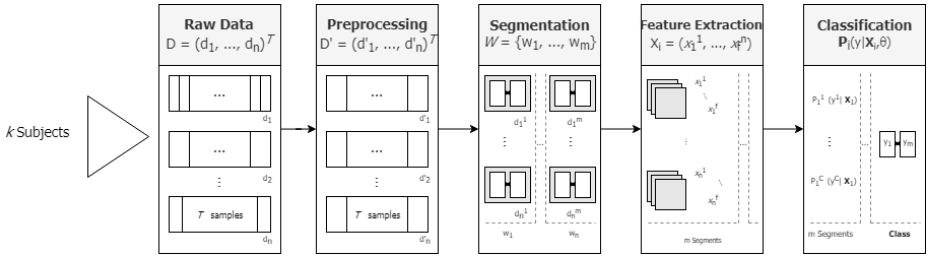

Figure 2.1: The Activity Recognition Chain, adapted from Bulling et al. [BBS14]

2.1.1 Data collection

As described by Bulling et al. [BBS14], the first stage of any ARC pipeline is the acquisition of raw data by sensors attached to the body. This is used as input data for the system. State-of-the-art systems can also include external sensors, such as cameras, for a refinement of the data acquisition or for the generation of labeled data to be used as training examples. When working with accelerometer data, the use of multiple sensors is notable. This offers the chance of generating specifically useful features such as computing the angle between the acceleration vectors.

Sensors

The use of body-worn sensors for the collection of data is the current state-of-the-art methodology. In the recent literature, examples using accelerometers, gyroscopes, and electrocardiograms are noticeable. Commercially, GPS, pedometers, and cameras have been successfully used; examples can be seen in PokemonGO[1], Xbox Kinect[2], and Google Fit[3] [PKA+17][PNW12].

Accelerometers are the most widespread body-worn sensor. Their small size, low cost, light weight, and high battery-efficiency make them an optimal choice for studies on HAR. Accelerometers measure proper acceleration, that is the physical acceleration measured by an object [Rin12, p. 150]. An accelerometer experiencing free-fall would measure a proper acceleration of zero, and one resting on the surface of the Earth would register an upwards acceleration vector of $g \approx 9.81 m/s^2$. When used for HAR, tri-axial accelerometers are the preferred sensor, as they allow the generation of proven features; uni-axial accelerometers, however, have also been tested and report high result correlation to the state-of-the-art tri-axial accelerometers [VBD+12].

[1] https://www.pokemongo.com/en-us/
[2] https://support.xbox.com/en-US/xbox-360/accessories/kinect-sensor-setup
[3] https://www.google.com/fit

Sampling frequency

According to the Nyquist–Shannon sampling theorem, there is a correlation between continuous-time and discrete-time signals. This correlation allows the establishment of a discrete sampling rate in order to negate the loss of information when measuring a limited bandwidth continuous-time signal. As defined by Shannon [Sha98]:

> If a function $f(t)$ contains no frequencies higher than W cps[4], it is completely determined by giving its ordinates at a series of points spaced $1/2W$ seconds apart.

All voluntary human movements are contained below the $20Hz$ threshold [KNM+06]. So in order to fully measure human activity, the sampling frequency of a discrete-time measuring device, such as an accelerometer, needs to be twice the amount of the movement's frequency. By following $t = \frac{n}{2W}$, an upper bound is set at $40Hz$ required for the lossless measurement of any voluntary human movement.

2.1.2 Data Pre-processing

The second stage of the ARC consists of a common preparation across all the different sample sets. According to Bulling et al. [BBS14] frequent procedures "may involve calibration, unit conversion, normalization, resampling, synchronization, or signal-level fusion". Labeling the data for training the classifier is also done in the pre-processing step.

Synchronization

Each sensor samples at a specific rate, but it may dynamically vary its frequency, for example for power-saving. In the second stage, all sensors are synchronized and adapted to a single time frame. Moreover, sample sets from different subjects are also integrated in the common time frame. In order to synchronize the data, studies commonly include recognizable patterns such as shaking the sensors, or executing specific activities, such as clapping or jumping [Våg17, p. 57].

2.1.3 Data segmentation

In the third stage the dataset is divided according to each encountered activity. This process is defined by Bulling et al. [BBS14] as *spotting*: the identification of data segments likely to contain activities. Each data segment will include a timestamp marking the start and another one marking the end: $W_i = (t_1, t_2)$. The activity segment is then composed of all the data segments defining that activity. In HAR

[4] Cycles per second

studies, data segments are defined as windows, which are used to generate features for the classifier. Windows can either be fixed or dynamic in size.

Dynamic windows tend to encompass a whole single activity segment as a unique data segment. The difficulty of properly setting up dynamic windows comes from the pre-processing stage. The previous removal of artifacts and noise from the data facilitates this process. The inclusion of specific activities, such as shaking the sensors or clapping, also eases this process. According the Krishnan and Cook [KC14], dynamic windows offer potential results compared to static segments, but exclusively on binary-classification problems where the classifier is requested to answer a *true/false* question. Experimented by Kozina et al. [KLG11] it is possible to translate an activity classification problem to a binary question, by detecting activity intensities instead of specifically identifying the current activity.

Fixed-size windows are the widespread segmenting approach for HAR. As opposed to dynamic windowing, all data segments have a fixed amount of samples, and the distance between t_1 and t_2 is a constant and known value. Using fixed-size windows makes the classification task easier for the classifier. When assigning labels for the training set, windows including more than one activity may be either removed for clarity, or changed to *transition* labels [HT16, p. 48]. In successful previous HAR studies, window sizes typically range from a tenth of a second to several seconds, and depend on the activity to be recognized. Too short windows might not describe the activity accurately, and too long windows might include several activities that get discarded [LL13].

2.1.4 Feature Generation

In order for the HAR system to assign labels to a window, a classifier is used. As explained by Lara and Labrador, it would be nearly impossible for two signals representing the same activity to be identical [LL13]. Hence the need for applying feature extraction methodologies: "filtering relevant information and obtaining quantitative measures that allow signals to be compared". Feeding features to the classifier greatly reduces its requested workload, since the same information is represented with less data. For example, two simple yet useful features to represent a window of data are its mean and standard deviation, as used by Nakajima et al. [PFN06]. Roughly, if a $3s$ window sampled at a $100Hz$ were to be represented by those two features, the total amount of data used as input for the classifier would be 150 times lower, making feature extraction a powerful tool. Fewer inputs for the classifier reduce training and testing times, and allow faster iterations when setting up a HAR system, making it attractive for studies such as this thesis. Noisy sets can also be cleaned up by using feature selection, which reduces overfitting from the classifier.

Two types of statistical features are commonly seen in studies focusing on HAR. These are time-domain features and frequency-domain features [PGKH08]. Time-domain refers to properties that describe the data over time, equivalent to the raw signal; as opposed to frequency-domain, where the data is described in relation to the repetition of its characteristics. Besides statistical features, HAR studies use other features, for example Structural features. As described by Olszewski [Ols01], structural features encompass complimentary information on the subject, such as its heart rate, which can help identify the ongoing activity.

Time-domain features:

A type of statistical features, time-domain features are extracted directly from the sensor data. The generation process includes a data collection step followed by all the calculations described for each feature. The data window can then be expressed as a single feature, or as a collection of features to be fed to the classifier.

Time-domain features can be calculated directly with the raw data, and are, therefore, inexpensive to generate. According to Khan et al. [KTKL14] and [KSL13], accuracies of over 90% can be achieved with the use of these features, while still benefiting from low energy consumption by the sensor system and the activity recognizer. The time-domain features used throughout this thesis are listed and described in Table 2.1.

Frequency-domain features:

Another type of statistical features, frequency-domain features are extracted from a frequency-domain transformation of the sensor data. By expressing movement as a repetition of acceleration vectors through time, periodic characteristics of movement can be noticed. Frequency-domain features allow the extrapolation of the repetitiveness from a movement, thus approaching the nature of the movement. As an example, the action of walking can be similarly described across different people if reduced to the repetitive task of lifting and advancing one foot after the other. The frequency-domain features used throughout this thesis are listed and described in Table 2.2.

In order to transform a raw signal to a frequency, Fourier analysis (FA) is employed. FA decomposes a function and represents it as an addition of oscillatory components. A visualization of the process is depicted in Figure 2.2.

Name	Definition	Description
Mean	$\tilde{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$	Arithmetic mean of values for an axis.
Standard deviation	$s_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \tilde{x}^2)}$	Root of the uncorrected variance (the average squared distance from the mean).
Skewness	$b_x = \frac{\frac{1}{n}\sum_{i=1}^{n}(x-\tilde{x}^3)}{s_x^3}$	How "skewed" the distribution of values are around the mean.
Magnitude max, mean, and SD	$m_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$	The maximum, mean, and standard deviation of the magnitude of the signal.
Correlation	$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\tilde{x})(y_i-\tilde{y})}{(n-1)s_x s_y}$	Pearson's product-moment coefficient. The degree of linear dependence between two series.
Zero cross rate	$zcr_x = \frac{\sum_{i=2}^{n}\lvert sgn(x_i)-sgn(x_{i-1})\rvert}{2(n-1)}$	Number of times the signal's value changes from negative to positive and vice versa.
Mean cross rate	$mcr_x = \frac{\sum_{i=2}^{n}\lvert sgn(x_i-\tilde{x})-sgn(x_{i-1}-\tilde{x})\rvert}{2(n-1)}$	Number of times the signal's value changes from over to under the mean and vice versa.
Root square mean	$\tilde{x} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$	The root of the mean of the squared values.
Energy	$E_x = \sqrt{\sum_{i=1}^{n}(x_i - \tilde{x})^2}$	A measure of the signal's strength.
Range	$max(x) - min(x)$	Difference between maximum and minimum of a sequence.

Table 2.1: Time-domain features

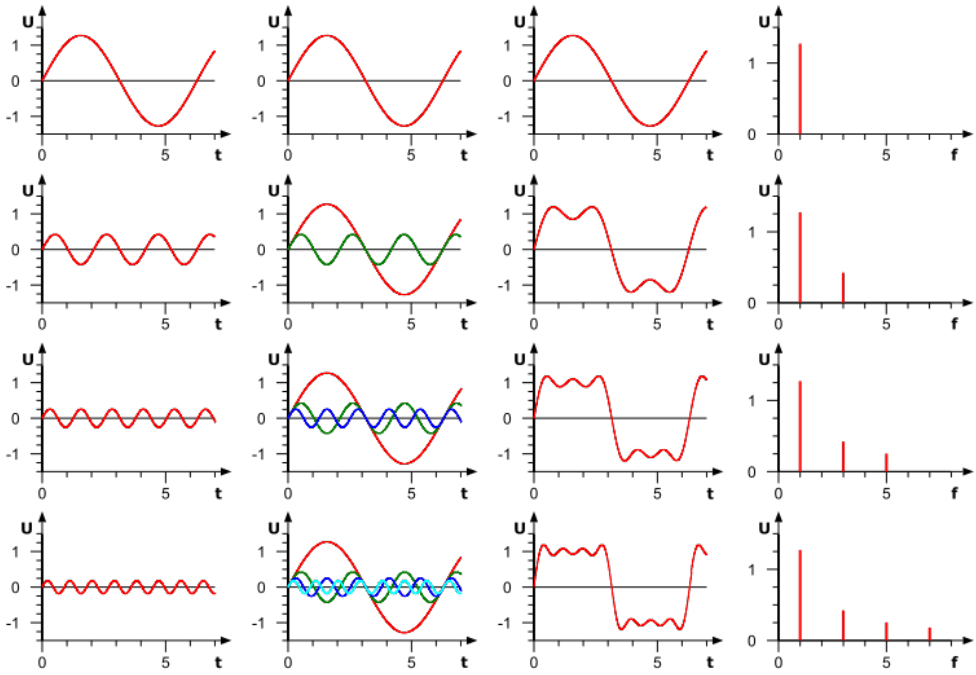

Figure 2.2: Fourier analysis of a period square wave. Each row adds a new periodic function. The second column superimposes them; the third column adds them; and the last column shows the amplitude of each periodic function. *Source: https://commons.wikimedia.org/wiki/File:Fourier_synthesis.svg*

2.1.5 Classification

The last stage is where an activity label is assigned to each window. The classification task. Commonly used in HAR systems are supervised learning algorithms. In supervised learning, previously labeled data is used as training data for a classifier, which, once trained, will be capable of labeling previously unseen data. Throughout the literature focusing on HAR, multiple types of supervised machine learning solutions are noticeable. Mainly: Support Vector Machines, k-Nearest Neighbors, Random Forests, and Artificial Neural Networks, among others [PNW12, KTKL14, KNM+06]. The implemented classifier for this thesis is a Random Forest classifier, which is explained in detail in the following section.

Name	Definition	Description
Mean amplitude	$\tilde{a} = \frac{1}{k}\sum_{j=0}^{k} a_i$	The arithmetic mean of the amplitudes.
Amplitude standard deviation	$s_a = \sqrt{\frac{1}{k}\sum_{j=0}^{k}(a_i - \tilde{a})^2}$	The root of the uncorrected variance for all the amplitudes.
Maximum amplitude	$max(a)$	The maximum amplitude.
Spectral centroid	$sc_a = \frac{\sum_{j=0}^{k} a_j \times f_j}{\sum_{j=0}^{k} a_j}$	Analogous to the center of mass of the frequencies if one regards the amplitude a_j as analogous to volume and the frequency f_j as analogous to density.
Dominant frequency	$f_(argmax_j a)$	The frequency with the maximum amplitude.
Spectral entropy	$p_j = \frac{a_j^2}{\sum_{j=0}^{k} a_j^2}$ $H = -\sum_{j=0}^{k} p_j log(p_j)$	The disorder in the spectrum.

Table 2.2: Frequency-domain features

2.2 Machine learning

An agent is considered to be learning if its performance on a task improves after making observations about the world [RN16]. Specifically in machine learning, the agent is trained with input-output pairs, and will, afterwards, be tested by determining the output of new inputs.

Supervised learning:

In supervised learning the agent must fabricate a function that maps inputs to outputs by observing a number of given input-output pairs. As defined by Russell and Norvig [RN16] the task is the following:

> Given a training set of N example input–output pairs
>
> $$(x_1, y_1), (x_2, y_2), ..., (x_N, y_N),$$
>
> where each y_j was generated by an unknown function $y = f(x)$, discover a function h that approximates the true function f.

The before mentioned input-output pairs correspond to (x_i, y_i) pairs. The function h is a hypothesis function that will be tested by using a test set. The test set is composed of examples distinct from those in the training set, and the test will yield the accuracy of the function h. If the learning problem outputs a single value from a finite set of options, it's called a classification problem. In HAR, the system is asked for a label that identifies an activity (walking, standing, sitting, among others) and is, therefore, a classification problem.

2.2.1 Random Forest

Random Forest is an ensemble learning classification model. Ensemble learning refers to the usage of multiple learning algorithms for the prediction of the classifier. Random forests work as large collections of single decision trees bagged together to form a forest. The bagging algorithm is outlined in Algorithm 2.1; each individual tree is trained independently with a subset of the training data set.

Algorithm 2.1 Random forest - Tree bagging

for $b = 1$ to B **do**

Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.

Train a classification or regression tree f_b on X_b, Y_b.

end for

In order to output a single, common, decision, each tree's output is used to form a mean result of the forest, in the form of $output = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$. In classification problems, where the output is a value among a finite set of options, a majority vote indicates the output of the forest.

Decision trees:

A decision tree is the representation of a function that transforms a vector of attributes into a single output value [RN16]. If the value can only be one from a list of predetermined outputs the task of the decision tree is that of classifying the input. When there are only two classes, the task is called binary classification, the output will be either true or false. In HAR, the classes, or labels, tend to be activities such as running; and the attribute vectors tend to be features describing the human activity such as acceleration vectors.

A learned tree takes the form of a set of $if - then$ rules. In order to classify a new instance, the tree is traversed from the root node downwards until a leaf node is reached. Trees are composed of a single root node and branches emerging from it. The branches can then lead to another subtree that repeats the process.

Node-branch combinations can represent question-answer combinations, or attribute-value combinations. The last node on each branch line is called the leaf node, and represents the final classification, or output, of that decision tree.

The process of generating a Decision tree from a set of attribute-class pairs is exemplified by Figure 2.3, the decision tree obtained from Table 2.3. This example is adapted from [Qui86, p. 87], and illustrates the reasoning behind a golf player and whether or not he would play on a Sunday, based on the weather.

Outlook	Temperature	Humidity	Windy	Plays?
Rain	Hot	High	False	no
Rain	Hot	High	True	no
Overcast	Hot	High	False	yes
Sun	Mild	High	False	yes
Sun	Cold	Normal	False	yes
Sun	Cold	Normal	True	no
Overcast	Cold	Normal	True	yes
Rain	Mild	High	False	no
Rain	Cold	Normal	False	yes
Sun	Mild	Normal	False	yes
Rain	Mild	Normal	True	yes
Overcast	Mild	High	True	yes
Overcast	Hot	Normal	False	yes
Sun	Mild	High	True	no

Table 2.3: Weather observations and whether or not golf was played. Adapted from [Qui86].

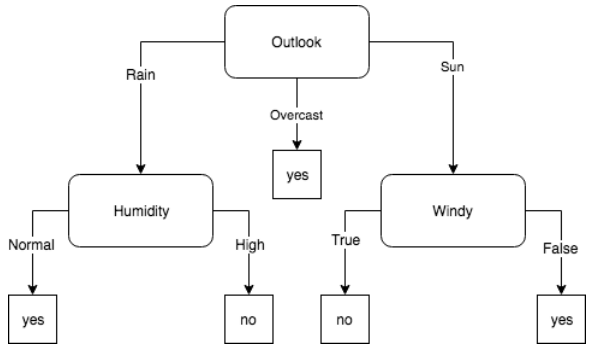

Figure 2.3: The decision tree resulting from training with the data in Table 2.3.

2.3 Quality metrics

This chapter has been explaining concepts as they were necessary throughout the logical implementation of a Human Activity Recognition system. After implementing, training, and testing the classifier, the system would be done per se. Metrics implanted to evaluate the performance of the system will now be introduced. These include indicators of how well the system performs in its predictions as well as which classes are more confusing for the system. One example of such metrics is the confusion matrix, explained in Table 2.4.

	Predicted class		
Actual class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Table 2.4: This table displays a confusion matrix of a binary classifier. True values are instances where both the predicted and actual class coincide: True Positives (TP) and True Negatives (TN). False values indicate wrong predictions: False Positives (FP) and False Negatives (FN). Positives mean that the prediction is Yes, and Negatives the opposite, No.

Accuracy is the percentage of correct predictions:

$$Accuracy = \frac{TP \cup TN}{TP \cup TN \cup FP \cup FN}$$

Precision denotes the percentage of actual positives among the total amount of predicted positives; it reflects how true is it when the system predicts a *Yes*:

$$Precision = \frac{TP}{TP \cup FP}$$

Recall denotes the percentage of correctly predicted positives among the total amount of positives; it reflects how likely the system is to predict *Yes* when the actual value is a *Yes*:

$$Recall = \frac{TP}{TP \cup FN}$$

F-Score is the harmonic average of the system's precision and recall, and is an estimate of how good the system is at performing its task:

$$F_1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In addition to the aforementioned metrics, classification systems make use of confusion matrices, such as the one in Table 2.5. Confusion matrices display the amount of classified instances for each single class, differentiating it between what it should and what it was classified as. This means that the diagonal going from the top-left to the bottom-right marks the true positives. Anything labeled outside this diagonal is an incorrect classification. Confusion matrices are 2×2 for binary classifiers, such as that in Table 2.4, or $n \times n$ where n is the finite amount of classes, such as that in Table 2.5. Confusion matrices are very useful when identifying difficult classes, since they clearly display the amount of instances correctly classified, as well as what was the output for incorrect classifications. For example, for the matrix displayed in Table 2.5 it would be concluded that the classifier struggles the most with the *standing* and the *walking* labels. In this example, it could be determined, for example, that the features describing the dataset should be tailored so that these two labels are better differentiated.

		Predicted				
Actual		standing	walking	bending	transition	sitting
	standing	3926	85	8	0	0
	walking	52	572	1	11	0
	bending	3	1	75	1	0
	transition	0	3	1	53	0
	sitting	0	0	0	4	14521

Table 2.5: This table displays a confusion matrix of a HAR classifier.

Chapter 3
Background and related work

A specialization project was conducted prior to this thesis where a variety of papers were studied. These consisted of several studies describing well performing HAR systems and well as studies on other classifiers using machine learning. The contents of this chapter are extracted and adapted from that project.

3.1 Motivation

In the process of starting a new study, there are always certain assumptions or facts that are taken from previous work. In some scenarios, limitations of the hardware used or the available resources set the bases of the research. In the specific scenario where a new study on human activity recognition takes place, a recurring lack of explanation on why a certain sampling rate of the accelerometer is used can be observed [MMH17]. The instinctive explanation may be one of the following, or a combination thereof: (a) the sampling rate was given by the accelerometer's capabilities; (b) the sampling rate was not the focus of the study, and therefore it wasn't experimented with; (c) the used sampling rate was taken from previous research. Explanations (a) and (b) were found to be most commonly used to explain a chosen sampling rate. Only two papers were found [LKK11, BKV+97] using a sampling rate based on previous research by Karantonis et al. [KNM+06]. The latter proved that all human activity was lower than *20Hz* and could therefore be captured by sampling at that frequency. The following questions started to appear: *Why is the choice of sampling rate not sufficiently justified in most HAR studies?* And: *What is an optimal sampling rate for HAR?*

3.2 Literature analysis

The objective of this research evolved as more papers were added to it. It started by searching studies on human activity recognition, looking for what sampling frequency was being used. Most papers were very centered on their machine-learning

implementations and lacked an explanation regarding the data they measured in general. Seeing this dynamic, the search shifted towards adaptive sampling techniques in sensor networks, not exclusively related to accelerometers nor activity recognition. There are some particular papers that are of interest to this project, given the techniques they implement for adaptive sampling.

3.2.1 Papers focused on Human Activity Recognition

Table 3.1 compiles 10 different papers that recognize human activity. They have very distinct sampling frequencies, ranging between $1Hz$ and $200Hz$, and most of them do not implement any adaptive sampling (AS). They will be evaluated one by one and their conclusions explained, as well as what can be extracted from them for this master's thesis:

Study reference	# channels	Data resolution	Frequency	Ad. Sampling
Kitchen HAR [MMH17]	6	-	64 Hz	-
Real-time HAR [KNM$^+$06]	3	-	100 Hz	-
HAR log-system [LKK11]	3	-	20 Hz	-
CNN HAR [LYC17]	3	-	1 Hz	Yes
Low-resolution HAR [KP08]	3	512 frames	-	-
Haar-like filtering HAR [HNK09]	3	-	200 Hz	Yes
Physical activity assessment [BKV$^+$97]	3	-	20 Hz	-
Daily activity classification [WWF11]	3	-	64 Hz	-
Pedometer [Zha10]	3	13 bits	50 Hz	Yes
Low-power fall detection [RZS12]	3	-	62.5 Hz	-

Table 3.1: Tri-axial accelerometer activity recognition examples

[MMH17] The study takes place in a closed environment, a kitchen. The aim is to prove that data-driven learners are more precise in identifying and labeling data than learners with handcrafted features. The study falls into the category of human activity recognition. The test subjects have two accelerometers attached to their bodies, one on each wrist. The data is collected and streamed at 64Hz. Mohammad's paper is related to the same fields than this thesis, and all the issues that have been described so far are present in it: *[Accelerometer data] were collected and streamed [...] at a frequency of 64Hz [MMH17].* There is a complete lack of explanation concerning their choice of sampling frequency, the stated extraction is all mention regarding the acquisition of their dataset.

[KNM+06] The study presents a real-time human movement classification system. The measuring device contains one accelerometer and a processor that identifies the current activity and transmits the label. The research cites other papers when reasoning why accelerometers are fit for activity recognition. The accelerometer samples at *100Hz* albeit activity recognition is effectuated using *45Hz*. Data is classified every second and then transmitted. This study is found to be cited repeatedly among the other studied papers, it sets the grounds for the two next papers when selecting a sampling rate: *All measured body movements are contained within frequency components below 20 Hz [KNM+06].* All conclusions and assumptions in the study are solid and explained, and set the groundwork for some of the other papers taken into account.

[LKK11] The study presents a personal-life-log containing an activity recognition implementation and an exercise information generator. The aim is to identify the activities correctly given a set of labeled data. One accelerometer at a frequency of *20Hz* is used. The sampling rate is extracted from [KNM+06]. A sliding window of 10s with 50% overlapping is used to compress the data before transmitting. All assumptions are extracted from very influential and solid studies, and make this paper a reliable source of methodologies that were applied to this thesis.

[LYC17] The study presents a one-dimensional convolutional neural network for human activity recognition. A single accelerometer is employed, gathering data at *1Hz*. This data is then processed and the magnitude vector is sent over the network. This study is notably recent, and outperforms implementations seen in other papers. Moreover, the amount of gathered data ranges between $\frac{1}{60}$ to $\frac{1}{300}$ compared to that from all the other studied papers. This indicates that the implemented features allow the usage of a lower total amount of samples, and still maintain a high precision metric.

[KP08] The study carries out HAR from low-resolution sensory streams. Accelerometers are used, the quantity is not specified, but it can be extracted from the data samples that they have one on each ankle. The sampling frequency is not specified. The sampled data is grouped every 512 samples and the average is transmitted over the network. This study proposes a sliding window with 50% overlap. There appears to be a complete disregard when it comes to explaining the data acquisition, this paper falls into the same category as [MMH17].

[HNK09] The study proposes a Haar-like[1] filtering technique in order to reduce computation costs when recognizing human activity. The data of a single accelerom-

[1] Haar-like features group identified inputs in a single simplified output. They are used to reduce the computation cost in face-recognition algorithms.

eter is sampled at *200Hz*. The data is processed in 50% overlapping windows of 512 frames before being transmitted. The objective of this study is lowering computation costs, which they relate to raw data analysis. The sampling frequency used is the highest of all papers reviewed, and there is a lack of explanation behind it.

[BKV+97] The study builds an accelerometer and data processing unit for the assessment of daily physical activity. Sampling frequencies are thoroughly discussed using previous studies. Their implementation consists of a low-pass at *0.11Hz* and a high-pass at *20Hz*, anything outside that range is not measured. This paper accurately links voluntary movements and the required sampling frequency to detect them, and sets the basis for adaptive sampling techniques that depend on the previously registered task.

[WWF11] The study proposes a multi-layered method for labeling human daily activity. Multiple parameters extracted from the accelerometer's data are used. A single accelerometer sampling at *64Hz* is employed. The interest in this paper is the data manipulation after communication, where extra information can be obtained from the same samples, without any additional measurements, by feature extraction.

[Zha10] The study implements a pedometer using a single accelerometer. The transmitted data is the average of every 50 samples. They determine that a step takes between 0.2s and 2s, and therefore they want to sample the average of every second. The chosen sampling rate is *50Hz*. This paper is very specific in their activity to be recognized, and makes more assumptions than the rest of the reviewed papers. The attractiveness of the paper comes from the implementation of non-overlapping windows where the average of every second is transmitted to be analyzed.

[RZS12] The study proposes an energy-efficient real-time fall detection system. It employs a single accelerometer sampling at a frequency of *1kHz*. The data analysis is effectuated every 16ms, a rate of *62.5Hz*. This paper proposes energy-efficient methods for solving their problem. Data transmission is reserved for the labels, not the raw data. The study recognizes and tackles the same problem dealt with in this thesis, with the purpose of lowering the energy consumption. As opposed to previous papers, it identifies the current activity on the spot, thus lowering the amount of transmitted data.

3.2.2 Papers focused on Adaptive Sampling techniques

As opposed to the previous studies, the following table [Table 3.2] compiles 3 papers that employ different methods driven by efficiency. This procedure is much more similar to the proposed method in this thesis, and the approach to the task will, therefore, be analyzed.

Study reference	Frequency	Ad. Sampling technique
IoT sensors data reduction [FKL17]	-	Weighted sequence selection
Data stream sampling [ESCD$^+$18]	0.2 Hz - 0.03 Hz	Averaged sampling window
Environmental parameter sensing [EHM16]	1.1 mHz	Conditional sampling window

Table 3.2: Adaptive sampling examples

[FKL17] The study thoroughly explores different methods for reducing outgoing data in sensor networks. The aim is to decrease the amount of stored data and the energy usage of the wireless transmitters. The paper tackles a multi-tier reduction mechanism, decreasing the outgoing information from the sensor as well as the incoming information to the routers. The research methodology and the problem they identify are in line with the objectives described for this thesis and will be followed during the methodology.

[ESCD$^+$18] The study aims to reduce the amount of used storage for gathered data. It proposes an implementation of Adaptive Sampling (AS). This paper is the culmination of a three part research comparing three different AS techniques. This paper is quoted during this thesis' result validation, as creating quantifiable results from each system to be compared is a relatively novel task.

[EHM16] The study implements and tests a new AS method for sensor networks. A procedure for recovering lost data is also described. The research focuses all transmission efforts on newly gathered data and assumes that only change is sent. The proposed system uses features that inherit from this paper, as described in the theoretical section.

Chapter 4
Methodology

This chapter serves as a prelude for the experiments run for this thesis. Two distinct sections are detailed. First, the implementation of this Master's thesis' HAR system will be explained using the ARC displayed in Figure 2.1. Second, the experiments' specifics will be described and reasoned, so that Chapter 5 can be limited to the execution and results.

4.1 Implemented system

As stated in Section 1.2, the objective of this thesis is that of comparing how a HAR system is affected by lowering its sampling rate. In order to do this, a data set will be obtained, processed, and analyzed with our implemented classifier. This process will be run several times with different sampling rates for the dataset and the results compared, in order to fulfill the objective.

This thesis does not try to give a singular answer to the question: *What is the best sampling rate to use when doing accelerometer-based HAR?* The correct answer to that question depends on the system requirements and constraints. As an example, medical studies might prefer higher precision values over battery life; whereas consumer applications using smart-wearables might be satisfied with lower precisions if the battery consumption remains within a determined limit. Therefore, to answer the question a guideline will be provided where each implementation is compared in terms of battery usage and attained precision. This comparison is going to define a set of optimal implementations within each value range, a Pareto optimality.

Pareto efficiency:

Pareto efficiency, or Pareto optimality, is defined by Teich [Tei01] as the set of all Global-optima points within a function. That means that one of the values cannot be optimized without affecting the other negatively. The red line in Figure 4.1

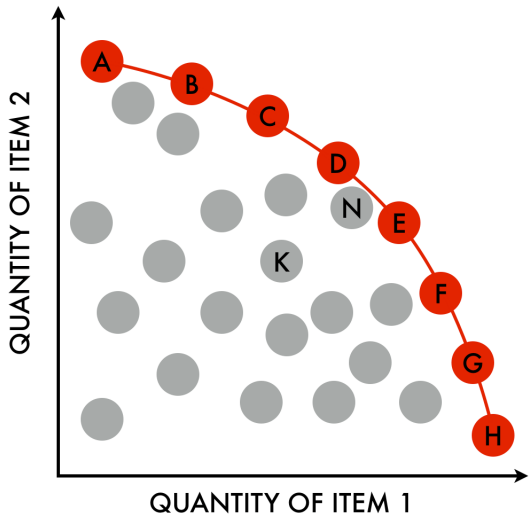

Figure 4.1: The red line denotes the Pareto front, where any point is the optimal value of one quantity in relation to the other. *Source: https://en.wikipedia.org/wiki/File:Pareto_Efficient_Frontier_1024x1024.png*

exemplifies this concept. In the figure, point K is a suboptimal solution, as there are values with higher quantities of both item 1 and 2, but any values between A and H have specific combinations that make them desirable depending on the system needs. Point H, for example, is the instance with the highest quantity of item 1.

4.1.1 Data Acquisition

The first step in the Activity Recognition Chain is the acquisition of raw data. This thesis builds upon the project of Hessen and Tessen [HT16] and has access to their same dataset. All the specifications of the sensors are taken from their project:

The data was collected using two tri-axial accelerometers. The devices were AX3 Axivity sensors[1]. The sampling rate was specified to be $100Hz$, and their locations were the upper back and the front-right thigh, as depicted in Figure 4.2. The dataset includes over 30 hours of labeled accelerometer data, a total of 10,901,356 readings at $100Hz$, from 35 different subjects. In order to label the data, the subjects were recorded and each sample was manually classified. A total of 19 labels were identified; the amount of instances, as well as their names can be seen in Table 4.1.

[1] https://axivity.com/product/ax3

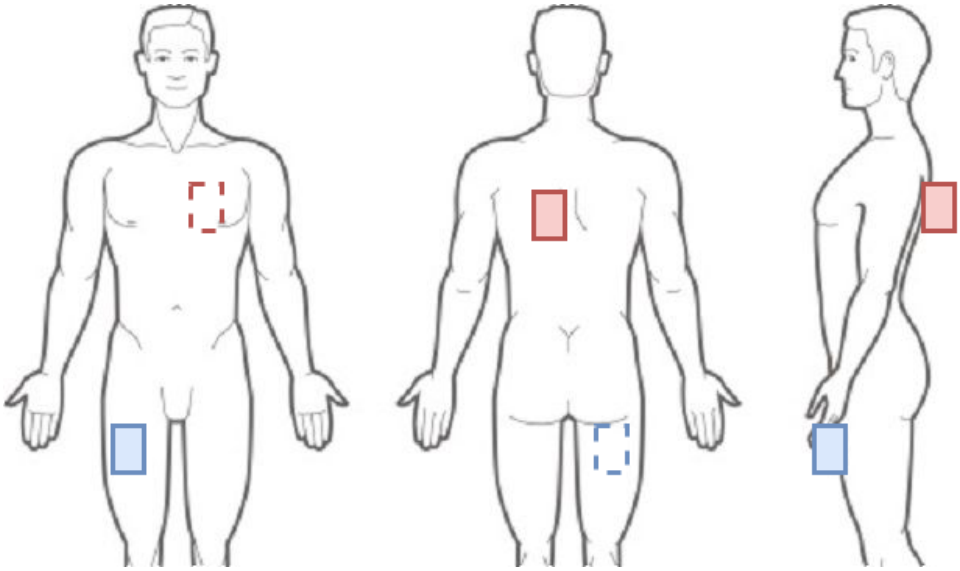

Figure 4.2: Representation of the location of the sensors. The red square marks the position of the upper back accelerometer, and the blue square marks the position of the right thigh accelerometer. *Source: https://www.pinterest.com/pin/489907265694622075/*

Given that the dataset was provided by another project, the data protection and anonymization processes extend from Hessen and Tessen [HT16, p. 35]. This project never had access to any data usable to identify the original subjects.

4.1.2 Pre-processing

After acquiring the data, it must be synchronized across all sensors and prepared to be segmented. Of the 19 labels described in Table 4.1, some were deleted and others joined together. Hessen and Tessen [HT16] built the first system and Vågeskar [Våg17] improved upon it, so both projects have been used as precedent when building this HAR system. The reasoning behind each specific class as well as the implications will now be exposed:

Deleted labels:

Worrisome labels are those considered to be non-representative of any relevant activities, those that are not specific enough to identify real movement, as well as those that are specific to the gathering of data. From the original 19 labels, a total of 6 have been removed, as was done in the referenced projects [HT16, Våg17].

Label name	Label amount	Label relative amount
Walking	1,274,404	11.69%
Running	93,499	0.86%
Shuffling	474,894	4.36%
Stairs (ascending)	103,676	0.95%
Stairs (descending)	91,193	0.84%
Standing	1,320,604	12.11%
Sitting	5,749,760	52.74%
Lying	653,808	6.00%
Transition	90,576	0.83%
Bending	41,149	0.38%
Picking	17,312	0.16%
Undefined	101,188	0.93%
Cycling (sitting)	535,926	4.92%
Cycling (standing)	48,676	0.45%
Heel-Drop	24	0.00%
Vigorous Activities	14,329	0.13%
Non-Vigorous Activities	57,230	0.52%
Commute (standing)	54,272	0.50%
Transport (sitting)	178,836	1.64%

Table 4.1: All the labels originally identified as well as the amount of instances of each, and the percentage relative to the total amount of labeled samples. The *Commute* and *Transport* labels were not described in Hessen and Tessen [HT16] but included in their dataset.

Heel Drop was removed from the dataset altogether. This activity was exclusively used to synchronize the sensors and was not part of any natural movement. A total of 24 samples were removed due to this.

Transitions were completely removed from the dataset. Transitions are defined as movements done between activities, such as getting on the bicycle or standing up from a chair. Transition activities do not necessarily relate to each other and may show little similarity. The inclusion of this label will affect the overall precision of our system, as it did for Hessen and Tessen [HT16], without giving any value to the results, since it won't be reproducible by future studies. A total of 90,576 samples were removed due to this.

Shuffling is defined as either a short walk or a stationary situation with leg movement. Shuffling overlaps with two other labels, but cannot be relabeled to any

specific one of those. Shuffling confuses the classifier without actually describing a specific action and was removed in both of the referenced projects [HT16, Våg17]. A total of 474,894 samples were removed due to this.

Samples labeled as either **undefined**, **non-vigorous activity**, and **vigorous activity** were all removed. None of these labels define a specific activity, and there are no patterns identifiable by the classifier, as shown by Hessen and Tessen [HT16, p. 49]. Similarly to the transition label, future studies will not be able to benefit from the inclusion of these labels and the performance of the classifier is unrealistically lowered, hence all those labels have been removed. The 101,188 undefined, 57,230 non-vigorous, as well as 14,329 vigorous samples were removed.

Renamed labels:

From the 13 labels remaining, 6 were renamed. This was done in order to either better represent reality, or make the results more comparable with other HAR studies.

Bending and **picking** were joined as a single label. More specifically, picking was renamed to bending, which was left the same. According to its definition, picking was identified when the subject grabbed an object from below the knee level. Bending is defined as bowing the torso downwards towards an object below knee level. The activity succession would, therefore, be Bending-Picking-Bending. Given the interclass similarity Hessen and Tessen [HT16] joined both classes together. After joining the classes, bending has a total of 58,461 samples.

Cycling (sit) and **Cycling (stand)** were both named **cycling**. Hessen and Tessen [HT16, p. 71] use only one label when comparing their classifier to the Acti4[2]. Even if the Cycling (sit) activity is often mislabeled as sitting, external HAR systems tend to use only one label to define both so, in order to give comparability to the implemented system, the same must be done.

Both **Commute (standing)** and **Transport (sitting)** were labeled according to what the subject was doing at that moment, excluding the context. Commute (standing) was relabeled to standing, and Transport (sitting) was relabeled to sitting. This renaming better represents what the user's actions reflect through their undergone acceleration.

Stairs (ascending) and **stairs (descending)** were relabeled to walking, as was done by Vågeskar [Våg17]. Walking up and down flights of stairs is very similar to walking on flat ground, and the classifier misclassifies those two classes as walking.

[2]https://www.ncbi.nlm.nih.gov/pubmed/25588819

After the preprocessing process, the total amount of samples represents over 28 hours of labeled data, with 10,163,115 individual samples at $100Hz$. The class disposition can be observed in Table 4.2. Note that running and Bending both have really low ($< 1\%$) amounts of data samples. This will affect the precision of the classifier, but removing those classes altogether would have affected the validity as well as the future relevance of this project, as those classes can potentially be interesting for other studies.

Name	Amount	Relative amount	Composition
Walking	1469273	14.46%	Walking + Stairs (ascending) + Stairs (descending)
Running	93499	0.92%	Running
Standing	1374876	13.53%	Standing + Commute (standing)
Sitting	5928596	58.33%	Sitting + Transport (sitting)
Lying	653,808	6.43%	Lying
Bending	58461	0.58%	Bending + Picking
Cycling	584602	5.75%	Cycling (sitting) + Cycling (standing)

Table 4.2: All the labels as they were used by the project. Displays amount of samples, relative amount, and labels included in each class.

4.1.3 Segmentation

The segmentation process consists of the formation of windows, grouping data instances, so that features can be extracted in the future. Windows work well in HAR since human activities are essentially repetitive and those repetitions can be encompassed in each window, naturally describing the activity. Throughout HAR literature window sizes range between $0.01s$ and $10s$ [LL13]. Hessen and Tessen [HT16, p. 52] use a window size of $1s$, and Vågeskar [Våg17] $3s$ windows.

Window size will not be determined beforehand for this project, as it would go against the planned objective. The proposed method includes a comparison between sampling rates and their effects depending on the window size of the tested scenario. Specifically, given that the dataset used was captured at a $100Hz$, and the target minimum sampling rate is $1Hz$, the window size will range from $1s$ to $10s$. If the results at $1Hz$ are precise enough to advocate an even lower sampling rate, the new tests will reach up to $0.1Hz$, maintaining the same window size.

The experiments section includes tests where windows were not used. In these, the classifier is fed the raw data without being segmented. The objective of this is testing the effects of single sample test, where the classifier is trained and tested with

uncorrelated instances of labeled data. The benefit of not using windows is that a trained classifier could potentially reduce its sampling rate to the absolute minimum.

4.1.4 Feature Extraction

The used features are those depicted in Tables 2.1 and 2.2. These features are those most commonly occurring in the literature studied, as well as in the referenced projects [HT16, Våg17].

4.1.5 Classification

The implemented classifier is a 32 tree Random Forest classifier where data is divided into a 2/3 - 1/3 train-test sample size. The reason behind all these constrains is exposed in this subsection:

The initial intention for this Master's thesis was that of implementing both a Random Forest and a Convolutional Neural Netowork (CNN), but was deprecated because training times for the CNN were dramatically slower. As done by Vågeskar [Våg17], CNNs were tested, but training times took over 30 minutes. Given the nature of the study, and the fact that extensive combinations of systems will be trained and tested, it is not feasible to implement a classifier that takes thirty minutes to train. As an example: Implementing 10 different window sizes with 5 different strides and 50 different sampling rates, if run non-parallelly, would take $1250h$ to train, making CNNs unsuitable for our system design. Exclusively using Random Forests for HAR is done extensively in the literature [LL13], therefore the validity of this study remains. Random Forests benefit from lack of hyperparameters to be tuned, when opposed to CNNs, besides the amount of trees; and using them takes away plenty of effort commonly invested in setting up the classifier. Random Forests use ensemble methods to average the decisions of multiple trees; these trees are trained with random subsets of the data, so over-fitting can be noticeably reduced. For these two reasons Random Forests are the chosen classifier to be used.

Before presenting the bulk of experiments for the project, some tests were run in order to evaluate the parameters to use in the system's classifier. A small number of those tests run during the setup of the classifier led to confusion matrices similar to the one shown in Figure 4.3, where one or more classes were missing. This test was a very specific case, where only 4 of the total 7 classes occurred in the test set. This happens when the random division of the dataset leaves no instances of those classes in the testing subdivision. The cause is the low amount of cases certain classes have in Table 4.2, *running*, for example, is less than 1% of the total labels. Given this occurrence, the training-test division of the sample set is higher than that originally tested, and the proposed system uses two thirds of the data to train and one third to test the classifier. By doing this, the amount of times some classes are missing in the

		PREDICTED						
		b'standing'	b'walking'	b'bending'	b'sitting'			
ACTUAL	b'standing'	83	0	2	0			
	b'walking'	0	8	0	0			
	b'bending'	0	0	1	1			
	b'sitting'	0	0	0	100			

Figure 4.3: Confusion matrix of a test showing no instances of either *running*, *lying*, nor *cycling*.

test set is expected to decrease. Hessen and Tessen [HT16] also opted for a division of 2/3 - 1/3 train-test sample size, which means this division is also the optimal for the dataset used.

Commonly, machine learning literature recommends between 64 and 128 decision trees to form a random forest [OPB12]. Several test were run, comparing a forest formed by 10 trees, one with 30, and one with 50. All the experiments use the whole dataset described previously: A total of $28h$ of labeled data organized into 7 distinct classes. The individual samples are divided in 300-samples windows, $3s$, and expressed with the features displayed in Table 2.1. The resulting data was further divided, using 2/3 of it for the training phase, and the remaining 1/3 for testing the classifier. The confusion matrices are displayed in Figures 4.4, 4.5, and 4.6. A difference of 1% in the F_1 Score can be noted between the 50-tree forest and the 10-tree forest. Simultaneously, the test with the 30-tree forest outperformed both the 50-tree forest as well as the 10-tree forest, therefore it is concluded that the computational cost of having forests over 30 trees is not worth the marginal gain in precision. Following the guideline of having a forest of a size multiple of 2^n (16, 32, 64, 128, 256, etc.) the chosen Random Forest is composed of 32 trees.

```
                                        predicted
  actual      b'standing'   b'walking'   b'bending'   b'sitting'     b'lying'   b'cycling'   b'running'
 b'standing'          1220           84           13           76            9           19            1
 b'walking'             72         1233            6           83           25           18           16
 b'bending'              0            2            9            3            1            2            0
 b'sitting'            102          121           10         6410           54           17            7
   b'lying'             15           30            3           10          626            1            7
 b'cycling'              3           13            4            5            1          590            1
 b'running'              0            0            0            2            1            0           63
Category: b'standing' Precision: 0.8640226628895185 Recall: 0.8579465541490858 Balanced F Statistic: 0.8609738884968243
Overall: Precision: 0.7594886041265446  Recall: 0.857869167297662 Balanced F Statistic: 0.782938411581065
Category: b'walking' Precision: 0.8314227916385705 Recall: 0.8485891259463 18 Balanced F Statistic: 0.8399182561307902
Overall: Precision: 0.7594886041265446  Recall: 0.857869167297662 Balanced F Statistic: 0.782938411581065
Category: b'bending' Precision: 0.2 Recall: 0.5294117647058824 Balanced F Statistic: 0.2117647058823297
Overall: Precision: 0.7594886041265446  Recall: 0.857869167297662 Balanced F Statistic: 0.782938411581065
Category: b'sitting' Precision: 0.9728335103961148 Recall: 0.9537271239398899 Balanced F Statistic: 0.9631855747558227
Overall: Precision: 0.7594886041265446  Recall: 0.857869167297662 Balanced F Statistic: 0.782938411581065
Category: b'lying' Precision: 0.8730822873082287 Recall: 0.9046242774566474 Balanced F Statistic: 0.8885734563520228
Overall: Precision: 0.7594886041265446  Recall: 0.857869167297662 Balanced F Statistic: 0.782938411581065
Category: b'cycling' Precision: 0.9119010819165378 Recall: 0.9562398703403565 Balanced F Statistic: 0.9335443037974683
Overall: Precision: 0.7594886041265446  Recall: 0.857869167297662 Balanced F Statistic: 0.782938411581065
Category: b'running' Precision: 0.6631578947368421 Recall: 0.9545454545454546 Balanced F Statistic: 0.782608695652174
Overall: Precision: 0.7594886041265446  Recall: 0.857869167297662 Balanced F Statistic: 0.782938411581065
```

Figure 4.4: Test with a Forest made of 10 decision trees. F_1 Score == 78.29%

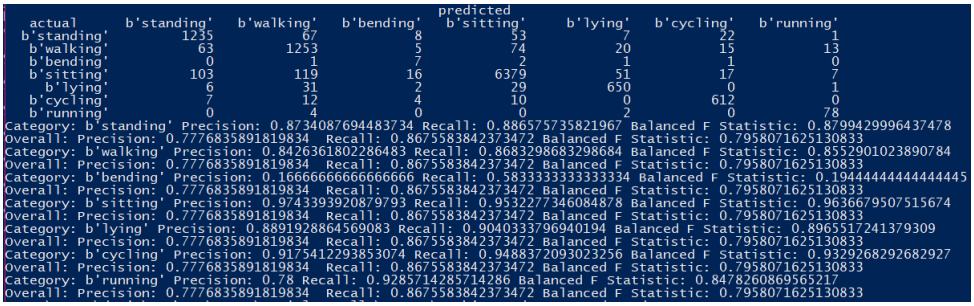

Figure 4.5: Test with a Forest made of 30 decision trees. F_1 Score == 79.58%

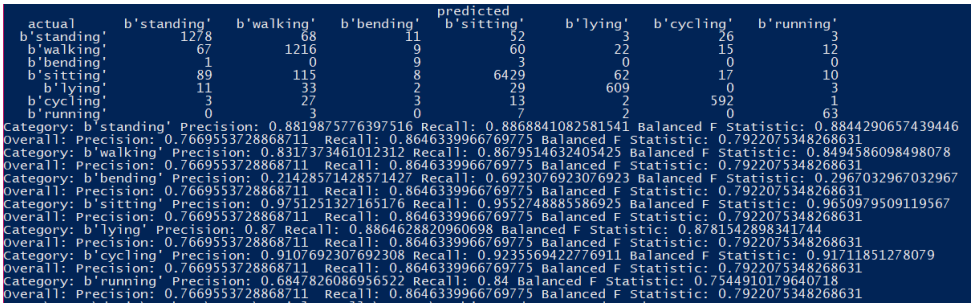

Figure 4.6: Test with a Forest made of 50 decision trees. F_1 Score == 79.22%

4.2 Experiments methodology

The experiments are divided in two sections: The first section makes no use of windowing/segmentation techniques, the classifier is fed the raw data, paired with a label, and asked to learn from that. The second section uses all the segmentation techniques and features described previously, the classifier is fed that feature combination paired with the label, as in the first section.

4.2.1 Objectives

The desired outcome of the experiments is a graphical comparison between the F_1 Score and the amount of samples used in each tested system. From this comparison, the Pareto Front can be obtained, fulfilling the main objective of this Master's thesis.

When asking the classifier for a prediction given a single instance of data, without windowing techniques, the sampling rate of the test does not appear to be relevant for the precision of the prediction. This is the hypothesis behind dividing the experiments

between windowed and non-windowed. If this were correct, a classifier could be trained with extensive amounts of data and, when needed, only a minimum amount of sampling time would be needed. The reason this happens is that the training process uses single data-frame inputs for training, and those do not depend on the sampling rate. The only variable that affect the classifier's performance in this scenario is the actual amount of single data samples used to train the system. The more samples, the better performance the system will have. The first set of experiments will prove this hypothesis, if correct.

The experiments with windows vary three parameters and compare the effect on the F_1 Score of the overall system, obtained by calculating the mean of the F_1 Score for each label. Calculating the overall F_1 as described, does not correctly describe the experiment, but allows the results to be compared between tests, and serves as a proper guideline if the system had different amounts of values for each class. The three parameters are: Window size, Window stride, and Sampling rate.

In order to reduce the sampling rate, a specific amount of samples will be eliminated. For example, if the desired sampling rate is $99Hz$ and the original is $100Hz$, every 99th sample will be eliminated. Following this logic, if the target sampling rate is $50Hz$, every second sample will be eliminated.

4.2.2 Experiments

This subsection covers every experiment necessary for achieving this Master's thesis goals. Detailed explanation on the constraints and variables is offered, as well as the objective behind running each specific experiment. The execution and results are explored in Chapter 5. The discussion and conclusion of the experiments is exposed in Chapter 6, where each implementation is compared so that future studies can make use of this project.

Experiment:	1.1 - Random parts of the complete dataset; no windows
Description:	From the 28h of labeled data, ten subsets of a lowered sampling rate will be generated. The sampling rates will be: $100Hz$, $90Hz$, $80Hz$, $70Hz$, $60Hz$, $50Hz$, $40Hz$, $30Hz$, $20Hz$, and $10Hz$. The generated subsets will be shuffled and the first 1,000,000 samples used for the experiment. This leaves 666,666 samples for training and 333,334 samples for testing each system. The result of this test will be one confusion matrix per implementation. This 7x7 matrix will display all the labels in the system, their occurrences and how they were classified. Besides the confusion matrix, the precision, recall, and F_1 Statistic for each category will be obtained. The overall precision, recall, and F_1 Statistic will also be given.
Constraints:	- Random Forest classifier - 32 trees - No windows used - 1,000,000 random samples per test - 66% - 33% Train - Test sample set division
Variables:	- The sampling rate will be lowered from $100Hz$ to $10Hz$. - The actual samples will be randomly taken from the global dataset.
Objectives:	The objective of this experiment is proving that, when trained with the same amount of data samples, lowering the sampling rate will not affect the performance of the classifier. This is exclusive to the current case, as no windowing techniques are involved.
Expected results:	Since all the implemented classifiers will be trained with the exact same amount of data, the results for each subsequent implementation should not differ in terms of attained precision. Besides differences originating from the randomness of the experiment, such as the test set containing more or less instances of a complicated label (*bending* or *running*).

Experiment:	1.2 - Fixed training, reduced testing; no windows
Description:	From the 28h of labeled data, one single training set will be extracted. A random selection of 666,666 samples destined for training. Besides the training set, 100 different testing sets will also be extracted from the remaining (excluding the ones directed for training) data instances. The random forest will be trained with the same dataset for each instance of a testing set. After training, the classifier will be tested with a subset of the global dataset, this subset will have a maximum of 333,333 samples (at a $100Hz$), and a minimum of 3,333 samples (at $1Hz$). The result of this test will be one confusion matrix per implementation. This 7x7 matrix will display all the labels in the system, their occurrences and how they were classified. Besides the confusion matrix, the precision, recall, and F_1 Statistic for each category will be obtained. The overall precision, recall, and F_1 Statistic will also be given.
Constraints:	- Random Forest classifier - 32 trees - No windows used - 666,666 fixed samples for training
Variables:	- Between 333,333 and 3,333 random samples for testing, ranging from $100Hz$ to $1Hz$. - The sampling rate on the testing set will be lowered from $100Hz$ to $1Hz$. - The actual testing samples will be randomly taken from the global dataset.
Objectives:	The objective of this experiment is testing what should be the minimum amount of samples (sampling rate) in the testing phase so that the classifier is reliable in it's predictions.
Expected results:	There will be a certain sampling frequency where the classifier is as precise as with the highest amount of testing samples. The initial expectation is around the $5Hz$ mark (16,665 samples). This low sampling rate could be used to accurately describe the performance of a classifier that does not use windows.

Experiment:	1.3 - Reduced testing - proof; no windows
Description:	This experiment continues from the previous one. From the 28h of labeled data, ten subsets will be extracted, ranging between 6,666,666 samples to 666,666 samples. The testing set will be the same size as the one in the previous experiment's conclusion; same size can mean both the same sampling rate as well as the exact same amount of samples, both cases will be tested. This means that each training set will have two test sets: one with the resulting amount from the previous experiment, and one with the relative size of the training set according to the resulting sampling rate from the previous experiment. The random forest will be trained with each training set, and tested twice, once for each test set. The result of this test will be one confusion matrix per implementation.
Constraints:	- Random Forest classifier - 32 trees - No windows used - A fixed size for the test set extracted from the previous experiment. - A fixed percentage for the test set, from the previous experiment.
Variables:	- The amount of samples on each subset will increase from 666,666 to 6,666,666. - The actual training samples will be randomly taken from the global dataset.
Objectives:	The objective of this experiment is testing the validity of the previous experiment when working with bigger datasets, as well as obtaining a value indicating how many samples are needed to correctly assess the current activity in windowless implementations.
Expected results:	There will be a specific amount of samples in the test set where the performance of the classifier can be analyzed. This amount will be dependant on the actual precision of the classifier, the amount of instances of each label in the global dataset, as well as the amount of labels. The actual value is expected to be the same as in the previous experiment.

Experiment:	1.4 - Fixed test, reduce training; no windows
Description:	From the 28h of labeled data, one hundred training sets will be extracted. These will range from $100Hz$ to $1Hz$, equating to a range between 10,000,000 and 100,000 samples. The division between training and testing will be the result from Experiment 1.3, either a test set with a fixed size, or a specific percentage of the training set. The random forest will be trained with each training set, and then tested with the specific test set. The result of this test will be one confusion matrix per implementation.
Constraints:	- Random Forest classifier - 32 trees - No windows used - Either a fixed amount of samples is the test set, or a specific percentage, depending on the results from the previous experiment.
Variables:	- The amount of samples on the used set will be between 10,000,000 and 100,000. This equates to a reduced sampling rate from $100Hz$ to $1Hz$. - The actual training samples will be consecutively taken from the global dataset.
Objectives:	The objective of this experiment is determining the amount of training needed by a non-windowed classifier to perform accurately. The results to this experiment will be the answer to Research Question 1 (What is the minimum amount of training data required to have a dependable Random Forest?).
Expected results:	Following a Pareto distribution, there will be a front made up of all the implementations where the precision-sample amount is the best it can be. The expectation is that the lower the amount of samples the lower the precision, and at one point in particular, lowering the sampling rate drastically lowers the precision. The point we are most interested in the the one just before that, following the Pareto principle (where 20% of the efforts give 80% of the results).

Experiment:	2.1 - Different window sizes; with windows
Description:	From the 28h of labeled data, twenty different subsets will be created. These will represent the same data segmented together in windows. The window sizes will range from $0.5s$ to $10s$, equating to 50 samples and 1,000 samples per window. Each window will be represented with the generated features, not with the raw data. The random forest will be trained with 66% of each subset, and then tested with the remaining windows. The result of this test will be one confusion matrix per implementation.
Constraints:	- Random Forest classifier - 32 trees - Time-domain and frequency-domain features used. - The full 28h of data will be used for each test. - 66% - 33% Train - Test sample set division
Variables:	- The size of the windows will range from $0.5s$ to $10s$. - The size of the subsets will range between 50 samples and 1,000 samples per window. Meaning that the test with the longest windows will have less data instances, even if the original data is the same.
Objectives:	The objective of this experiment is determining the optimal window size given our system's constraints. After finishing this experiment, a specific result will be set up as our benchmark to reference.
Expected results:	Given the previous projects using the same dataset and techniques, the expected result is that the system using 3s windows has a good balance between amount of data instances and amount of data per instance, giving the best results. The expectations for this experiment are that a single, or a very short range of, window size outperforms the the other cases, and can be set as our highest precision system to compare to.

Experiment:	2.2 - Reduce amount of samples in windows; training and testing
Description:	Continuing from the previous experiment, and using the window size established in Experiment 2.1, in this experiment the sampling rate in both the training and testing set will be lowered. From the 28h of data 100 different subsets will be extracted and segmented. These will range from $100Hz$ to $1Hz$, with a minimum amount of 2 samples per window. The window size will be the result from experiment 2.1. It is possible that certain features, such as the *Zero Cross Rate* contribute no information to the system, the used features might be evaluated depending on the results. Features to reevaluate are: *zero cross rate*, *mean cross rate*, and *energy*. The result of this test will be one confusion matrix per implementation.
Constraints:	- Random Forest classifier - 32 trees - Time-domain features used. - The window size that offered the best results in Experiment 2.1 will be used for all tests. - The global data set will be used for each test. - 66% - 33% Train - Test sample set division
Variables:	- The sampling rate used for the experiment will range from $100Hz$ to $1Hz$.
Objectives:	The objective of this experiment is determining the optimal sampling rate for consecutive windows, as well as extending the previous experiment to include a data reduction in the training phase.
Expected results:	Depending on whether or not a windowless classifier performs within decent standards at low sampling rates, it is possible that implementing windows is counterproductive precision-wise. The expectation is that some reduction in the sampling rate is possible, but windowed implementations are more susceptible to extremely low sampling rates.

Experiment:	2.3 - Reduce amount of windows; training and testing
Description:	Using the complete global dataset, multiple subsets will be extracted. These will cover all the possibilities between varying the window size (seconds worth of samples), the window stride (seconds between windows), and the window density (sampling rate within the window). Window sizes will range between 0.5 and 10 seconds; window stride will range between 0.5 and 100 seconds; window density will range between the original and 2 samples per window. If it was determined, on the previous experiments, that window density should not be lower than a specific amount, that will be the minimum sampling rate for each window. The result of this test will be one confusion matrix per implementation. The implementations will be compared to one another by the collective amount of samples used in the subset.
Constraints:	- Random Forest classifier - 32 trees - Time-domain and frequency-domain features used. - The global data set will be used for each test. - 66% - 33% Train - Test sample set division
Variables:	- The sampling rate used for the experiment will range from $100Hz$ to the specified amount in Experiment 2.3. - The distance between windows will range between 0.5 and 100 seconds. - Window size will range between 0.5 and 10 second windows.
Objectives:	The objective of this experiment is answering Research Question 2 (How does the usage of windowing techniques affect the precision of Random Forests?). With this experiment, a comparison between all possible implementations will be obtained.
Expected results:	The expected result is that a balanced implementation with a slightly lower sampling rate (such as $20Hz$), a moderately lengthier window (5s, for example) and a window stride depending on the window size (e.g. the same size as the window) will have the best performance.

Experiment:	2.4 - Variable parameters in testing; best performing systems
Description:	In this experiment the parameters used for training and for testing will not be the same. The tested variables are window size, window stride and window density, applied separately to the testing set. The training sets and their parameters will be those in the Pareto front obtained in the previous experiment. Each implementation that performed well in the previous experiment will be tested with multiple datasets, to try and lower the amount of data used when testing the system. The result of this test will be one confusion matrix per implementation. The implementations will be compared to one another by the collective amount of samples used in the subset.
Constraints:	- Random Forest classifier - 32 trees - Time-domain and frequency-domain features used. - All the sets in the Pareto front of the previous experiment will be used. - 66% - 33% Train - Test sample set division
Variables:	- The sampling rate used for the test set will range from $100Hz$ to the specified amount in Experiment 2.3. - The distance between windows in the test set will range between 0.5 and 100 seconds. - Window size in the test set will range between 0.5 and 10 second windows.
Objectives:	The last experiment addresses the hypothesis that the techniques explored in this project do not need to be constant between training and testing. The whole system, including the training phase and the testing phase, can implement different variables to reduce the overall amount of gathered data. The objective, therefore, is exploring whether or not the system can perform within acceptable limits and still reduce the total amount of collected data, as well as determining the relationship between the efforts in gathering and classifying the data.
Expected results:	The expected result is that activities tend to last up to minutes, so skipping measurements won't harm the overall precision, while still helping with the amount of gathered data. The prediction also includes that as much data as possible should be gathered for training, but testing only needs a fraction of that: generating (as in training) a HAR system will take more effort than using it, but it's usage will consume almost no power in comparison.

Chapter 5
Experiments

5.1 Experiment 1.1

In experiment 1.1 ten different tests were conducted. From the original sample set, ten randomized subsets were extracted. These contained 1,000,000 samples each. The sampling rate varied from $100Hz$ to $10Hz$, that is of every 100 consecutive samples only a fraction were used. For example, in the $20Hz$ case, 20 out of 100 samples were used, effectively reducing the usable samples by 80%. After the reduction, 1,000,000 random samples were extracted, resulting in ten sets of the same size. All the test use a 32 tree RF with 2/3 - 1/3 train-test sample division as classifier.

Results:

The hypothesis was that the tests' performances would be similar, since the quality of the training was the same. Quality refers to having the same amount of samples dedicated to training in each test. The hypothesis has been proven correct, as seen in Figure 5.1 and 5.2. The tables represent the confusion matrices from the $100Hz$ case and the $10Hz$ case, which should, if the hypothesis were incorrect, have the highest result disparity. As depicted, the results are the same. Figures 5.3 and 5.4 show the complete set of quality metrics of those two same examples. Figure 5.5 compares all the quality metrics, where a clear flat tendency is illustrated.

Conclusion:

When the RF is trained with a similar windowless dataset the only metric that affects the system's performance is the amount of samples used for training, not its sampling rate. Similar refers to the amount and quality of data instances, as in the number of samples the forest trained with, and what those samples represented. For example, training a RF with 90,000 samples of sitting and 10,000 samples of all the other labels, as opposed to 100,000 samples of properly distributed data instances across all the system's classes will not yield the same performance.

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	758	298	2	7	185	228	478	1956	0,59%
	b'cycling'	48	16055	6	32	319	314	2178	18952	5,74%
	b'lying'	6	74	19987	5	466	126	672	21336	6,47%
	b'running'	3	148	25	1631	89	93	1082	3071	0,93%
	b'sitting'	25	865	217	35	187994	1119	2113	192368	58,29%
	b'standing'	40	517	78	37	875	38084	4513	44144	13,38%
	b'walking'	84	1908	148	262	1354	4768	39649	48173	14,60%
	TOTAL	964	19865	20463	2009	191282	44732	50685	330000	100,00%
	PERCENTAGE	0,29%	6,02%	6,20%	0,61%	57,96%	13,56%	15,36%	100,00%	

Figure 5.1: Experiment 1.1: Confusion matrix of the $100Hz$ system. *No. trees = 32; Train samples = 667,000; Test samples = 330,000*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	745	249	0	10	172	219	481	1876	0,57%
	b'cycling'	55	16129	10	14	318	350	2124	19000	5,76%
	b'lying'	13	72	19968	10	460	110	655	21288	6,45%
	b'running'	4	153	15	1580	71	102	1073	2998	0,91%
	b'sitting'	27	911	221	39	188088	1213	2038	192537	58,34%
	b'standing'	47	510	65	39	884	38110	4642	44297	13,42%
	b'walking'	94	1774	174	287	1399	4651	39625	48004	14,55%
	TOTAL	985	19798	20453	1979	191392	44755	50638	330000	100,00%
	PERCENTAGE	0,30%	6,00%	6,20%	0,60%	58,00%	13,56%	15,34%	100,00%	

Figure 5.2: Experiment 1.1: Confusion matrix of the $10Hz$ system. *No. trees = 32; Train samples = 667,000; Test samples = 330,000*

								MACRO	
PRECISION	78,63%	80,82%	97,67%	81,18%	98,28%	85,14%	78,23%	85,71%	PRECISION
RECALL	38,75%	84,71%	93,68%	53,11%	97,73%	86,27%	82,31%	76,65%	RECALL
F-1 SCORE	51,92%	82,72%	95,63%	64,21%	98,00%	85,70%	80,21%	80,93%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	92,17%	ACCURACY

Figure 5.3: Experiment 1.1: Quality metrics of the $100Hz$ system. *No. trees = 32; Train samples = 667,000; Test samples = 330,000*

								MACRO	
PRECISION	75,63%	81,47%	97,63%	79,84%	98,27%	85,15%	78,25%	85,18%	PRECISION
RECALL	39,71%	84,89%	93,80%	52,70%	97,69%	86,03%	82,55%	76,77%	RECALL
F-1 SCORE	52,08%	83,14%	95,68%	63,49%	97,98%	85,59%	80,34%	80,75%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	92,20%	ACCURACY

Figure 5.4: Experiment 1.1: Quality metrics of the $10Hz$ system. *No. trees = 32; Train samples = 667,000; Test samples = 330,000*

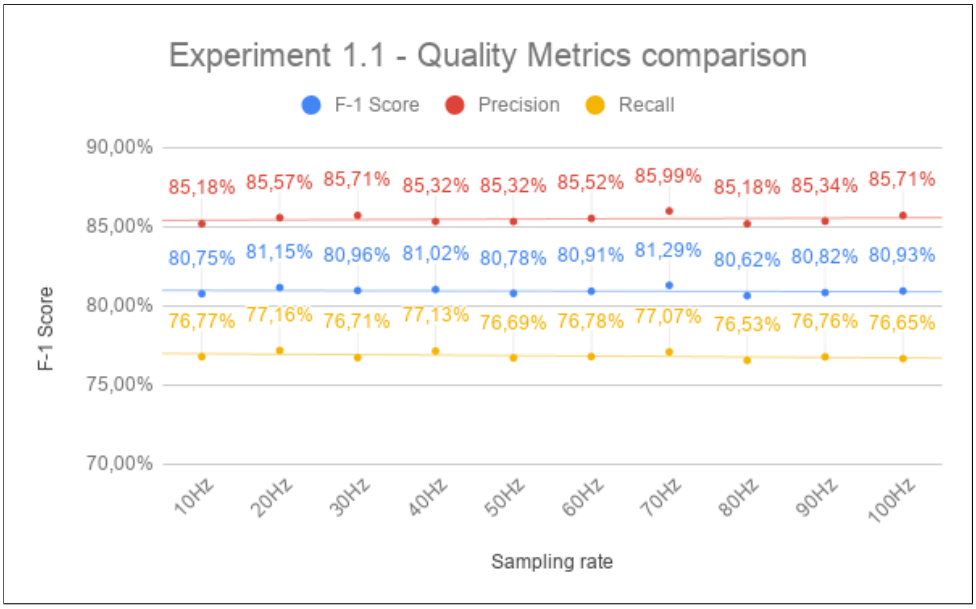

Figure 5.5: Experiment 1.1: Quality metrics comparison of experiment 1.1. Even with different sampling rates, all systems used the same amount of total (training and testing) samples. *No. trees = 32; Train samples = 667,000; Test samples = 330,000*

5.2 Experiment 1.2

Experiment 1.2 is composed of 100 different tests. In these, one specific training set was tested with 100 different test sets of diminishing size. The training set was composed of 666,666 samples. The test sets start at 333,333 samples, representing $100Hz$, and end at 3,333 samples, representing $1Hz$. The RF was trained with the training set and then tested with each test set, with the intention of discovering what size should the test set be in respect of the train set.

Results:

Depicted in Figure 5.6 is the confusion matrix for the test set containing 333,333 samples ($100Hz$). Figure 5.7 represents the $1Hz$ set. Notable differences between the confusion matrices are the amount of samples in general, specifically in the *bending* category. There are 1921 instances of *bending* in the first table, as opposed to 14 instances in the second. *Bending* represents 0.58% of the global dataset, and the exact same amount in the $100Hz$ test. On the $1Hz$ test set, however, *bending*

represents 0.42% of the samples, that is over 27% fewer samples that represent the action of *bending*.

Figures 5.8 and 5.9 illustrate the quality metrics for the $100Hz$ and $1Hz$ tests, respectively. The main notable differences are the lower scores on the metrics for the *bending* class, going from a 56% to a 23% F_1 Score.

Figure 5.10 represents the distribution of precision, recall, and F_1 score of each test for this experiment. It is distinguishable that the results are constant for all implementations over $11Hz$, but results with lower sampling rates have noticeable lower performance. There is also a tendency of results deviating more from the tendency value as sampling rates get lower, examples of such behaviour are the $1Hz$ and the $4Hz$ implementations.

Conclusion:

The results of this test are inconclusive, for the lowest tested sampling rate did not reach a failure state where it would be proven that higher and more specific sampling rates are necessary to test a given system. Instead, the tendency shown in Figure 5.10 appears to be that lower results are more random in their distribution and therefore, less reliable. This was not the aim of this experiment, since the wanted outcome was a relationship between the training set size and the required size of the test set. It can be concluded that a lower size collides with the frequency of the least represented class, which was mentioned in the expected results for experiment 1.3 and will, therefore, be explored more in depth in that experiment.

In order to conclude this experiment, an extension is proposed where the sampling rate is lowered even more, reaching $0.1Hz$, this experiment will be conducted in the following section.

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	830	245	5	8	174	224	435	1921	0,58%
	b'cycling'	37	16400	5	21	333	293	1922	19011	5,70%
	b'lying'	4	77	20301	18	440	114	652	21606	6,48%
	b'running'	5	142	25	1828	92	76	957	3125	0,94%
	b'sitting'	24	890	210	32	190009	1110	2040	194315	58,29%
	b'standing'	38	420	63	33	866	39290	4504	45214	13,56%
	b'walking'	79	1719	178	246	1279	4411	40229	48141	14,44%
	TOTAL	1017	19893	20787	2186	193193	45518	50739	333333	100,00%
	PERCENTAGE	0,31%	5,97%	6,24%	0,66%	57,96%	13,66%	15,22%	100,00%	

Figure 5.6: Experiment 1.2: Confusion matrix of the $100Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 333,333*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	2	2	0	1	2	3	4	14	0,42%
	b'cycling'	0	164	0	0	5	2	15	186	5,58%
	b'lying'	0	1	210	0	2	1	8	222	6,66%
	b'running'	0	3	1	22	0	1	7	34	1,02%
	b'sitting'	0	5	2	0	1898	16	21	1942	58,27%
	b'standing'	1	4	1	0	9	383	46	444	13,32%
	b'walking'	0	25	0	4	18	50	394	491	14,73%
	TOTAL	3	204	214	27	1934	456	495	3333	100,00%
	PERCENTAGE	0,09%	6,12%	6,42%	0,81%	58,03%	13,68%	14,85%	100,00%	

Figure 5.7: Experiment 1.2: Confusion matrix of the $1Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 3,333*

							MACRO	
PRECISION	81,61%	82,44%	97,66%	83,62%	98,35%	86,32%	79,29%	87,04%	PRECISION
RECALL	43,21%	86,27%	93,96%	58,50%	97,78%	86,90%	83,56%	78,60%	RECALL
F-1 SCORE	56,50%	84,31%	95,78%	68,84%	98,07%	86,61%	81,37%	81,64%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	92,67%	ACCURACY

Figure 5.8: Experiment 1.2: Quality metrics of the $100Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 333,333*

							MACRO	
PRECISION	66,67%	80,39%	98,13%	81,48%	98,14%	83,99%	79,60%	84,06%	PRECISION
RECALL	14,29%	88,17%	94,59%	64,71%	97,73%	86,26%	80,24%	75,14%	RECALL
F-1 SCORE	23,53%	84,10%	96,33%	72,13%	97,94%	85,11%	79,92%	77,01%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	92,20%	ACCURACY

Figure 5.9: Experiment 1.2: Quality metrics of the $1Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 3,333*

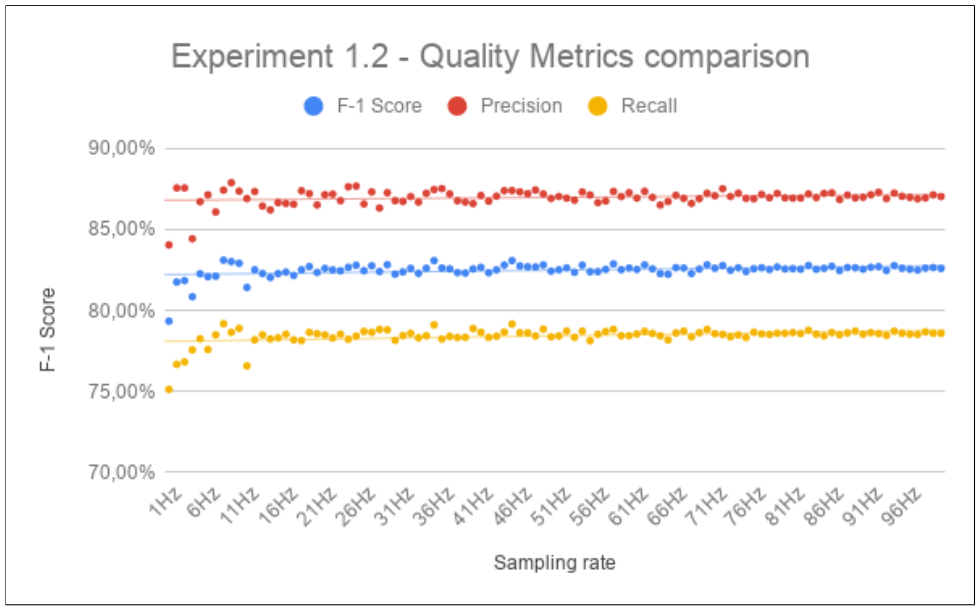

Figure 5.10: Experiment 1.2: Quality metrics comparison of experiment 1.2. Each implementation used the same training set and a reduced test set. *No. trees = 32; Train samples = 666,666; Test samples = (333,333 - 3,333)*

5.2.1 Extension of experiment 1.2:

Nine new sets were created and tested with the same training set of 666,666 samples. The new test sets represented sampling rates between $0.9Hz$ and $0.1Hz$, with steps of $0.1Hz$ each. Figures 5.11 and 5.12 show the confusion matrices of the $0.2Hz$ and $0.1Hz$ tests respectively. Figures 5.13 and 5.14 depict the quality metrics for those same two tests, and Figure 5.15 plots the comparison of all the quality metrics. This experiment's extension proves that sampling rates as low as the implemented ones are not suitable to test systems using this dataset, since the class imbalance makes the system's performance random. As shown in Figure 5.13, the $0.2Hz$ system is the best performer so far, that is because, with the low amount of test samples, no instances were mislabeled as bending, making its precision 100%. This high precision does not represent the system's actual precision. As seen in Figure 5.14, the $0.1Hz$ system is the worst performer, that is because there were no correct classifications of bending, so its precision is 0%. The difference between two test sets that are so similar in size makes test this small irrelevant for the system's performance. It is, therefore, concluded that future tests must have at least 10% of their dataset or 33,333 samples dedicated to testing, as these values mark the beginning of incongruent results and anything lower will randomly determine the performance of the classifier.

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	0	1	0	0	0	0	0	1	0,30%
	b'cycling'	0	13	0	0	0	0	3	16	4,80%
	b'lying'	0	0	19	0	1	0	1	21	6,31%
	b'running'	0	1	0	4	0	0	1	6	1,80%
	b'sitting'	0	1	0	0	184	1	4	190	57,06%
	b'standing'	0	1	0	0	1	41	4	47	14,11%
	b'walking'	0	3	0	2	1	5	41	52	15,62%
	TOTAL	0	20	19	6	187	47	54	333	100,00%
	PERCENTAGE	0,00%	6,01%	5,71%	1,80%	56,16%	14,11%	16,22%	100,00%	

Figure 5.11: Experiment 1.2 extended: Confusion matrix of the $0.1Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 333*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	1	0	0	0	1	0	0	2	0,30%
	b'cycling'	0	34	0	0	2	0	5	41	6,16%
	b'lying'	0	0	33	0	1	0	1	35	5,26%
	b'running'	0	0	0	6	0	0	3	9	1,35%
	b'sitting'	0	1	0	0	382	3	4	390	58,56%
	b'standing'	0	2	1	0	2	78	10	93	13,96%
	b'walking'	0	6	0	0	3	12	75	96	14,41%
	TOTAL	1	43	34	6	391	93	98	666	100,00%
	PERCENTAGE	0,15%	6,46%	5,11%	0,90%	58,71%	13,96%	14,71%	100,00%	

Figure 5.12: Experiment 1.2 extended: Confusion matrix of the $0.2Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 666*

							MACRO	
PRECISION	0,00%	65,00%	100,00%	66,67%	98,40%	87,23%	75,93%	70,46%	PRECISION
RECALL	0,00%	81,25%	90,48%	66,67%	96,84%	87,23%	78,85%	71,62%	RECALL
F-1 SCORE	0,00%	72,22%	95,00%	66,67%	97,61%	87,23%	77,36%	70,87%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	90,69%	ACCURACY

Figure 5.13: Experiment 1.2 extended: Quality metrics of the $0.1Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 333*

							MACRO	
PRECISION	100,00%	79,07%	97,06%	100,00%	97,70%	83,87%	76,53%	90,60%	PRECISION
RECALL	50,00%	82,93%	94,29%	66,67%	97,95%	83,87%	78,13%	79,12%	RECALL
F-1 SCORE	66,67%	80,95%	95,65%	80,00%	97,82%	83,87%	77,32%	83,18%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	91,44%	ACCURACY

Figure 5.14: Experiment 1.2 extended: Quality metrics of the $0.2Hz$ system. *No. trees = 32; Train samples = 666,666; Test samples = 666*

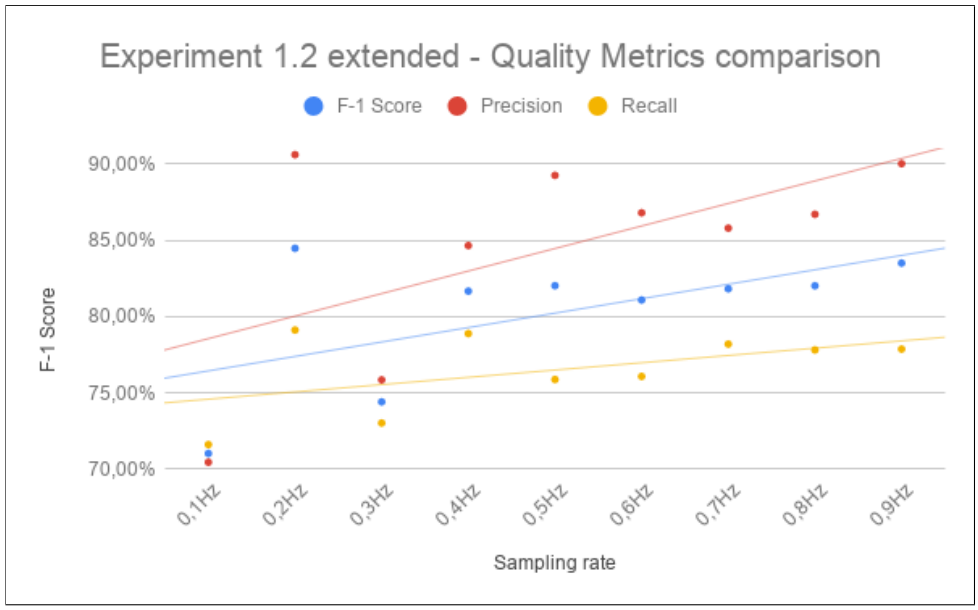

Figure 5.15: Experiment 1.2 extended: Quality metrics comparison of the extension of experiment 1.2. Each implementation used the same training set and a reduced test set. *No. trees = 32; Train samples = 666,666; Test samples = (333 - 2,999)*

5.3 Experiment 1.3

Experiment 1.3 puts the previous conclusion under the test. The previous assumption is that new test should have either 33,333 samples dedicated to testing or 10% of the total size of the subset. In this experiment, ten larger subsets were used, and, following the assumption, two test set were generated for each set. The subsets ranged between 666,666 and 6,666,666 samples; each set was trained and tested twice, once using a test set containing exactly 33,333 samples, and once using a test set containing 10% of the total amount of samples. For example, for the scenario containing 4,666,666 samples, one test used a 90% - 10% train - test division, and the other test a 99.29% - 0.71% division ($33,333/4,666,666 = 0.007124 \simeq 0.71\%$).

Results:

Figure 5.16 and Figure 5.17 show the confusion matrices of both tests done with the 4,666,666 set. Figure 5.16, illustrates the 10% case, where 466,667 samples were used for testing; Figure 5.17 depicts the other case, with 33,333 testing samples. Figure 5.18 portrays the quality metrics for the first test, where it can be seen that the system has an excellent performance compared to Figure 5.19, which represents

the 33,333 case, and displays lower scores for most metrics. The label that presents the worst results is the *bending* class, which has a 10% difference in precision and is directly responsible for the lower score in the metrics' average. Figure 5.20 shows the comparison of quality metrics throughout the experiment, no information can be obtained from this graph, except that lower amounts of samples have lower performance, which was previously obtained knowledge. In opposition, Figure 5.21 compares and plots the difference between the quality metrics across the 10% and the 33,333 test for each set size. This table manages to represent a tendency towards a null gain the larger the dataset becomes.

Conclusion:

It is concluded that the 10% line is a good overall estimate, but with larger datasets (over 3,000,000 samples), reserving as large an amount of samples for testing has diminishing or null returns, as seen by the tendency in Figure 5.21. Therefore, when working with large datasets a fixed amount of test samples will be allocated, a visual representation of this conclusion would be the positive side of a Sigmoid function, which gains as the X grows but reaches a limit at some point. This experiment has not been enough to give a conclusion of small datasets, and will, therefore, be expanded upon further.

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	72	25	0	0	12	30	37	176	0,53%
	b'cycling'	5	1639	1	2	27	27	180	1881	5,64%
	b'lying'	1	12	2004	0	29	10	58	2114	6,34%
	b'running'	0	7	0	181	5	13	87	293	0,88%
	b'sitting'	3	72	21	3	19250	83	177	19609	58,83%
	b'standing'	4	36	7	1	85	3959	424	4516	13,55%
	b'walking'	11	183	11	28	106	371	4034	4744	14,23%
	TOTAL	96	1974	2044	215	19514	4493	4997	33333	100,00%
	PERCENTAGE	0,29%	5,92%	6,13%	0,65%	58,54%	13,48%	14,99%	100,00%	

Figure 5.16: Experiment 1.3: Confusion matrix of the 4,666,666 samples system. *No. trees = 32; Train samples = 4,633,333; Test samples = 33,333*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	1239	328	7	9	165	294	637	2679	0,57%
	b'cycling'	48	22955	6	31	404	367	2776	26587	5,70%
	b'lying'	3	106	28769	11	459	120	767	30235	6,48%
	b'running'	4	201	18	2613	76	98	1284	4294	0,92%
	b'sitting'	26	1016	266	53	267490	1252	2568	272671	58,43%
	b'standing'	29	553	96	58	891	55551	5884	63062	13,51%
	b'walking'	102	2362	146	359	1340	5466	57364	67139	14,39%
	TOTAL	1451	27521	29308	3134	270825	63148	71280	466667	100,00%
	PERCENTAGE	0,31%	5,90%	6,28%	0,67%	58,03%	13,53%	15,27%	100,00%	

Figure 5.17: Experiment 1.3: Confusion matrix of the 4,666,666 samples system. *No. trees = 32; Train samples = 4,199,999; Test samples = 466,667*

								MACRO	
PRECISION	75,00%	83,03%	98,04%	84,19%	98,65%	88,11%	80,73%	86,82%	PRECISION
RECALL	40,91%	87,13%	94,80%	61,77%	98,17%	87,67%	85,03%	79,35%	RECALL
F-1 SCORE	52,94%	85,03%	96,39%	71,26%	98,41%	87,89%	82,83%	82,92%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	93,42%	ACCURACY

Figure 5.18: Experiment 1.3: Quality metrics of the 4,666,666 samples system. *No. trees = 32; Train samples = 4,633,333; Test samples = 33,333*

								MACRO	
PRECISION	85,39%	83,41%	98,16%	83,38%	98,77%	87,97%	80,48%	88,22%	PRECISION
RECALL	46,25%	86,34%	95,15%	60,85%	98,10%	88,09%	85,44%	80,03%	RECALL
F-1 SCORE	60,00%	84,85%	96,63%	70,36%	98,43%	88,03%	82,88%	83,93%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	93,42%	ACCURACY

Figure 5.19: Experiment 1.3: Quality metrics of the 4,666,666 samples system. *No. trees = 32; Train samples = 4,199,999; Test samples = 466,667*

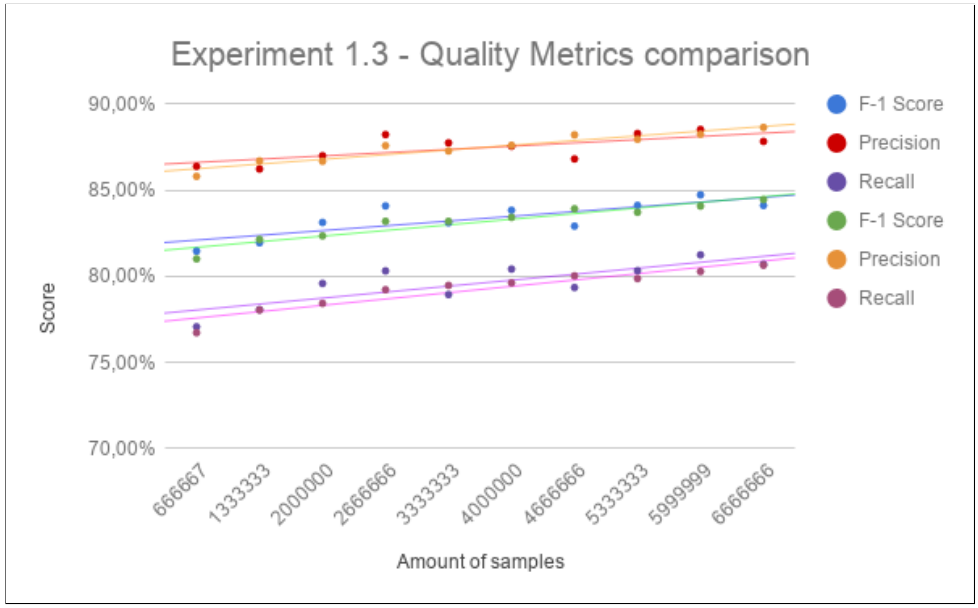

Figure 5.20: Experiment 1.3: Quality metrics comparison of experiment 1.3. Blue, red, and purple represent the 33,333 tests; green orange and magenta represent the 10% tests. *No. trees = 32; Train samples = (6,633,333 - 599,999); Test samples = (666,667 - 33,333)*

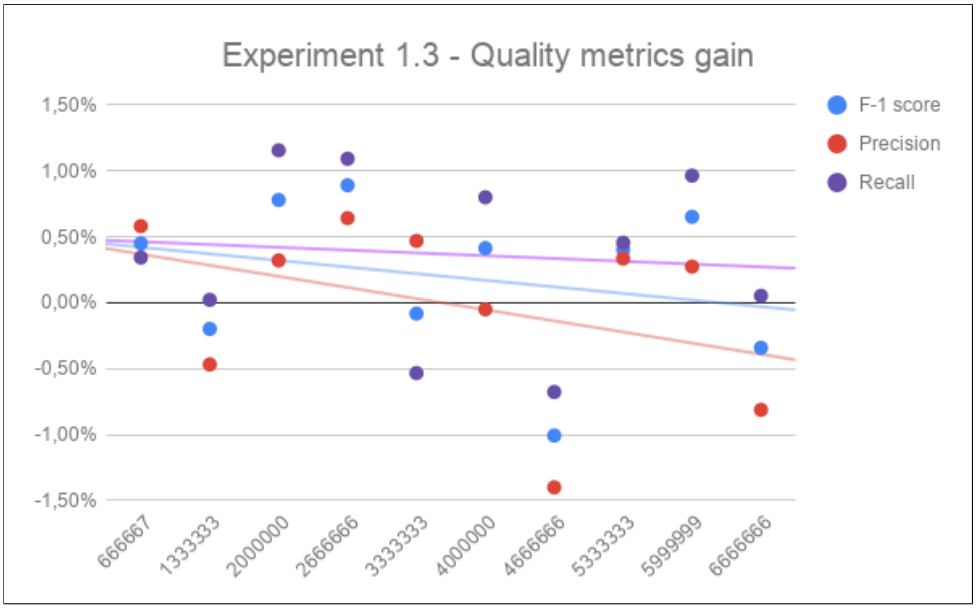

Figure 5.21: Experiment 1.3: Difference between the precision, recall, and F_1 score between the systems using the 33,333 samples and the 10% set size for testing.

5.3.1 Extension of experiment 1.3:

In order to understand the effect that setting a specific train - test division has on the performance of the classifier, it is imperative that the whole range of dataset sizes are explored. This extension amplifies experiment 1.3 and adds nine smaller subsets ranging between 66,666 and 600,000 samples, which represent between one and nine hundredth of the largest dataset in experiment 1.3 (6,666,666 data instances). This extension will test the same, using each subset twice, once employing 33,333 samples for testing and the other time allocating 10% of the total size for testing. For the 66,666 case, for example, the 33,333 test represents 50% of the total size, and the 10% case uses 6,666 samples.

Figures 5.22 and 5.23 show the confusion matrices of the 66,666 case. Comparing Figure 5.24 and Figure 5.25, which also illustrate the case using 66,666 samples, it is noticeable that the fixed test (where the test size was fixed to 33,333 samples) underperformed in most classifications, and has a remarkably lower recall for all labels. The quality metrics comparison and the gain in each category can be observed in Figures 5.26 and 5.27 respectively. It is visible that the usage of a fixed test set size has negative effects on the smaller datasets.

The final conclusion from this experiment is that specifying a fixed size for the dataset has negative results in the smaller datasets, slightly positive in the medium sets, and almost no positive outcome in the larger datasets. From this experiment it can be obtained that the worst results come when the fixed size represents more than 10% of the total size of the set. This is the result of leaving lower amount of data for the training phase, which leads to a classifier that performs poorly in comparison. Allocating between 5% and 15% of the dataset for testing and the rest for training appears to be the correct behaviour when conducting HAR.

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	58	29	0	0	25	37	48	197	0,59%
	b'cycling'	9	1471	2	0	58	49	260	1849	5,55%
	b'lying'	1	5	1944	3	90	16	81	2140	6,42%
	b'running'	1	19	6	120	15	5	127	293	0,88%
	b'sitting'	3	120	31	3	18917	160	264	19498	58,49%
	b'standing'	8	74	11	3	171	3760	474	4501	13,50%
	b'walking'	10	217	34	21	243	647	3683	4855	14,57%
	TOTAL	90	1935	2028	150	19519	4674	4937	33333	100,00%
	PERCENTAGE	0,27%	5,81%	6,08%	0,45%	58,56%	14,02%	14,81%	100,00%	

Figure 5.22: Experiment 1.3 extended: Confusion matrix of the 66,666 samples system. *No. trees = 32; Train samples = 33,333; Test samples = 33,333*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	17	3	0	0	6	5	9	40	0,60%
	b'cycling'	3	308	0	1	3	4	44	363	5,44%
	b'lying'	0	0	407	0	18	1	11	437	6,55%
	b'running'	0	6	2	33	3	1	24	69	1,03%
	b'sitting'	0	23	2	1	3766	26	47	3865	57,97%
	b'standing'	1	15	1	0	34	726	107	884	13,26%
	b'walking'	5	47	9	6	37	103	802	1009	15,13%
	TOTAL	26	402	421	41	3867	866	1044	6667	100,00%
	PERCENTAGE	0,39%	6,03%	6,31%	0,61%	58,00%	12,99%	15,66%	100,00%	

Figure 5.23: Experiment 1.3 extended: Confusion matrix of the 66,666 samples system. *No. trees = 32; Train samples = 59,999; Test samples = 6,667*

							MACRO	
PRECISION	64,44%	76,02%	95,86%	80,00%	96,92%	80,45%	74,60%	81,18%	PRECISION
RECALL	29,44%	79,56%	90,84%	40,96%	97,02%	83,54%	75,86%	71,03%	RECALL
F-1 SCORE	40,42%	77,75%	93,28%	54,18%	96,97%	81,96%	75,22%	75,77%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	89,86%	ACCURACY

Figure 5.24: Experiment 1.3 extended: Quality metrics of the 66,666 samples system. *No. trees = 32; Train samples = 33,333; Test samples = 33,333*

								MACRO	
PRECISION	65,38%	76,62%	96,67%	80,49%	97,39%	83,83%	76,82%	82,46%	PRECISION
RECALL	42,50%	84,85%	93,14%	47,83%	97,44%	82,13%	79,48%	75,34%	RECALL
F-1 SCORE	51,52%	80,52%	94,87%	60,00%	97,41%	82,97%	78,13%	78,74%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	90,88%	ACCURACY

Figure 5.25: Experiment 1.3 extended: Quality metrics of the 66,666 samples system. *No. trees = 32; Train samples = 59,999; Test samples = 6,667*

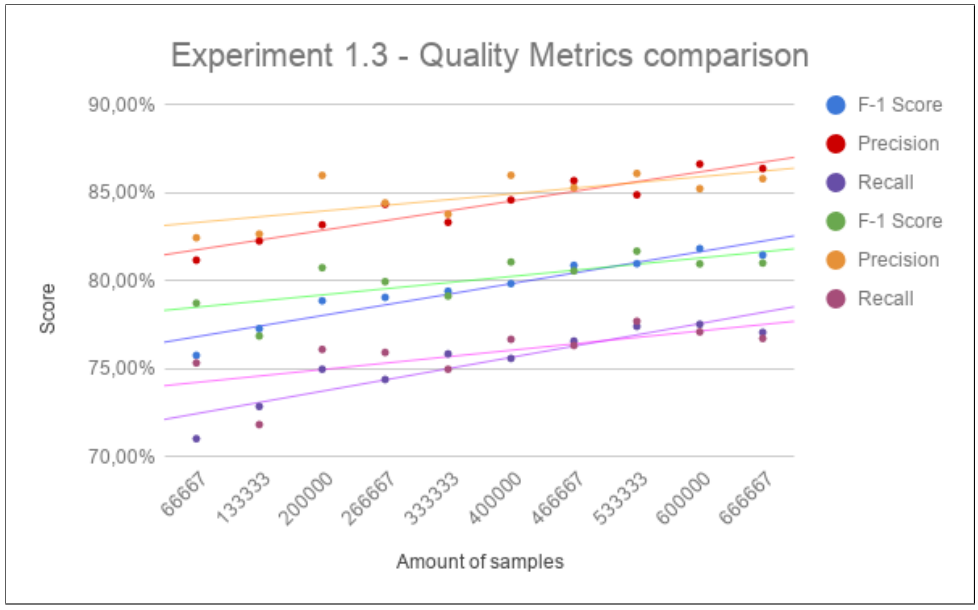

Figure 5.26: Experiment 1.3 extended: Quality metrics comparison of the extension of experiment 1.3. Blue, red, and purple represent the 33,333 tests; green orange and magenta represent the 10% tests. *No. trees = 32; Train samples = (566,665 - 59,999); Test samples = (66,666 - 6,667)*

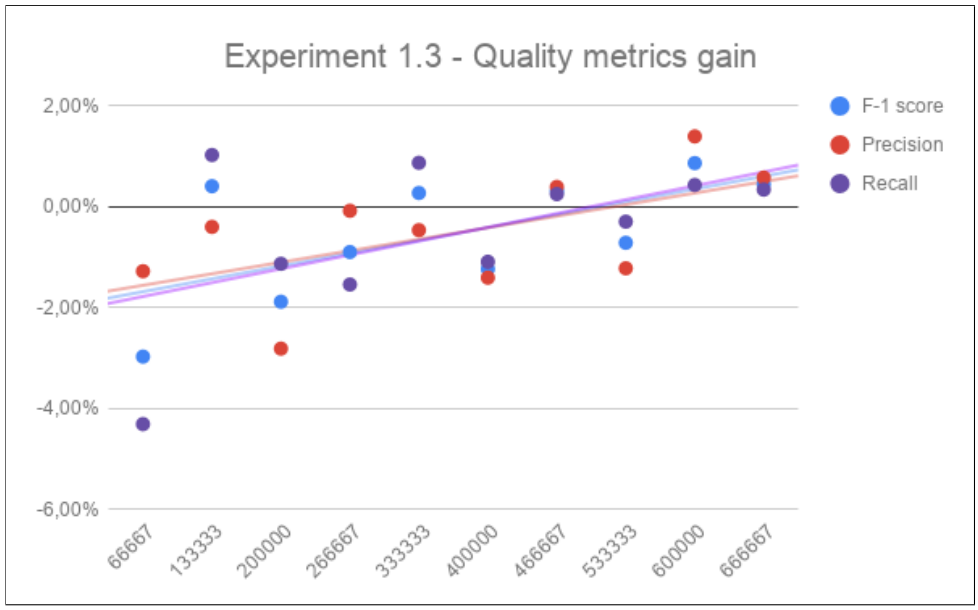

Figure 5.27: Experiment 1.3 extended: Difference between the precision, recall, and F_1 score between the systems using the 33,333 samples and the 10% set size for testing.

5.4 Experiment 1.4

Experiment 1.4 is the final experiment that does not implement windows, and compares all sampling rates tested until now and their performance. The tested sets range between $0.1Hz$ and $90Hz$, increasing by steps of $0.1Hz$ between the $0.1Hz$ and $1Hz$ values, and by steps of $1Hz$ between the $1Hz$ and $90Hz$ values. Values over $90Hz$ were not tested since the results flatten out after $50Hz$ and the computational times increase with the larger datasets. The test consists of a 32 tree RF classifier with 10,000,000 random samples from the original dataset, adapted to each given sampling rate. Given that the original dataset was recorded at $100Hz$, this reduction means that the $10Hz$ set will contain one out of every ten measurements, or 10% of the original dataset (for a total of 1,000,000 samples).

Results:

Figure 5.28 displays the comparison between the tests with sampling rates between $1Hz$ and $90Hz$. It can be noted that the results increase constantly up until the $70Hz$ mark, where they flatten and do not go over the 85% F_1 score mark. Figures 5.29 and 5.30 showcase the confusion matrix and quality metrics of the worst performer

in the first graph, the $1Hz$ system. In comparison, the $8Hz$ system shows some consistent results, as seen in Figures 5.31 and 5.32, where an F_1 score of over 80% is illustrated. Figure 5.33 shows the comparison between the tests with sampling rates between $0.1Hz$ and $1Hz$. These results are lower than any over $3Hz$, and are not as consistent with the neighboring tests, as can be seen in the overall comparison in Figure 5.34.

Conclusion:

This experiment proves the validity of single measurements without implementing windows, but warns about the randomness of the results when lower-than-usual frequencies are used. Consistent results come from using sampling rates over $5Hz$, which prove to be eighteen times more efficient than systems using $90Hz$ as their sampling frequency, and still maintain 94% of the performance of those. As seen in Figure 5.34, lowering the sampling rate under $2Hz$ can have anything between excellent and underwhelming performance, depending on the amount of samples found in the worse classes, which is a random factor.

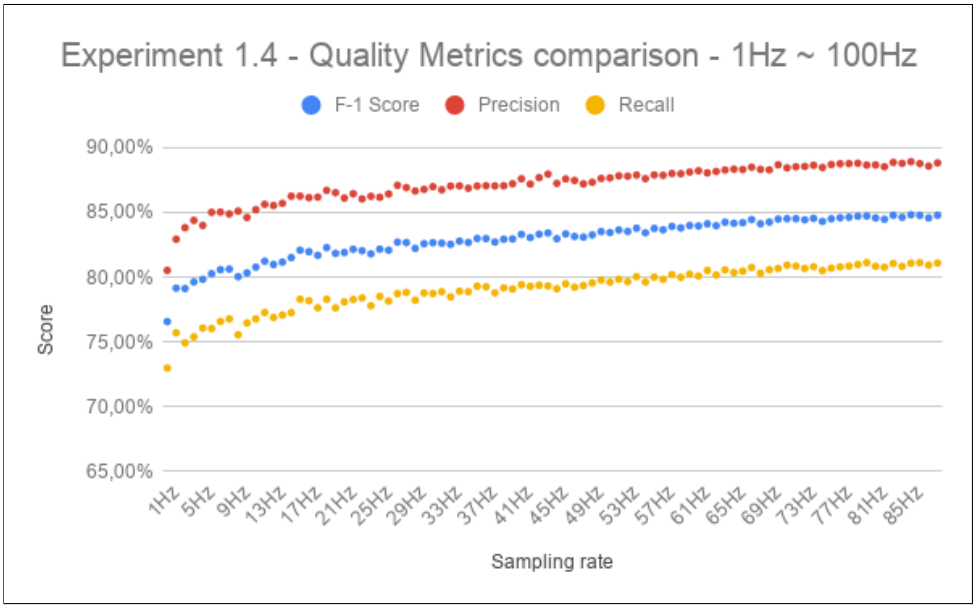

Figure 5.28: Experiment 1.4: Quality metrics comparison of the tests using set between $1Hz$ and $90Hz$. *No. trees = 32; Train samples = (8,100,000 - 90,000); Test samples = (900,000 - 10,000)*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	14	7	0	0	5	6	14	46	0,46%
	b'cycling'	7	497	0	0	7	12	67	590	5,90%
	b'lying'	0	1	590	0	24	2	20	637	6,37%
	b'running'	0	4	3	41	5	2	39	94	0,94%
	b'sitting'	1	26	9	2	5683	63	61	5845	58,45%
	b'standing'	3	16	0	1	44	1153	148	1365	13,65%
	b'walking'	2	68	3	8	47	180	1115	1423	14,23%
	TOTAL	27	619	605	52	5815	1418	1464	10000	100,00%
	PERCENTAGE	0,27%	6,19%	6,05%	0,52%	58,15%	14,18%	14,64%	100,00%	

Figure 5.29: Experiment 1.4: Confusion matrix of the $1Hz$ system. *No. trees = 32; Train samples = 90,000; Test samples = 10,000*

								MACRO	
PRECISION	51,85%	80,29%	97,52%	78,85%	97,73%	81,31%	76,16%	80,53%	PRECISION
RECALL	30,43%	84,24%	92,62%	43,62%	97,23%	84,47%	78,36%	72,99%	RECALL
F-1 SCORE	38,36%	82,22%	95,01%	56,16%	97,48%	82,86%	77,24%	76,58%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	90,93%	ACCURACY

Figure 5.30: Experiment 1.4: Quality metrics of the $1Hz$ system. *No. trees = 32; Train samples = 90,000; Test samples = 10,000*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	174	56	0	2	52	53	123	460	0,58%
	b'cycling'	15	3907	1	6	64	72	524	4589	5,74%
	b'lying'	0	20	4760	4	118	23	149	5074	6,34%
	b'running'	1	38	7	435	24	16	246	767	0,96%
	b'sitting'	4	274	48	10	45627	284	541	46788	58,49%
	b'standing'	12	127	29	9	199	9194	1243	10813	13,52%
	b'walking'	24	473	46	68	311	1209	9378	11509	14,39%
	TOTAL	230	4895	4891	534	46395	10851	12204	80000	100,00%
	PERCENTAGE	0,29%	6,12%	6,11%	0,67%	57,99%	13,56%	15,26%	100,00%	

Figure 5.31: Experiment 1.4: Confusion matrix of the $8Hz$ system. *No. trees = 32; Train samples = 720,000; Test samples = 80,000*

								MACRO	
PRECISION	75,65%	79,82%	97,32%	81,46%	98,34%	84,73%	76,84%	84,88%	PRECISION
RECALL	37,83%	85,14%	93,81%	56,71%	97,52%	85,03%	81,48%	76,79%	RECALL
F-1 SCORE	50,43%	82,39%	95,53%	66,87%	97,93%	84,88%	79,10%	80,63%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	91,84%	ACCURACY

Figure 5.32: Experiment 1.4: Quality metrics of the $8Hz$ system. *No. trees = 32; Train samples = 720,000; Test samples = 80,000*

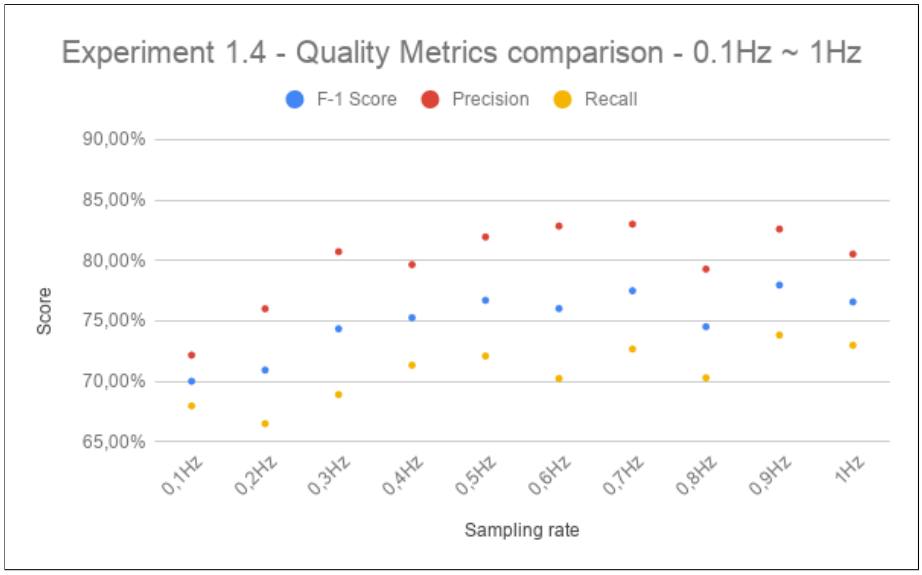

Figure 5.33: Experiment 1.4: Quality metrics comparison of the tests using set between $0.1Hz$ and $1Hz$. *No. trees = 32; Train samples = (90,000 - 9,000); Test samples = (10,000 - 1,000)*

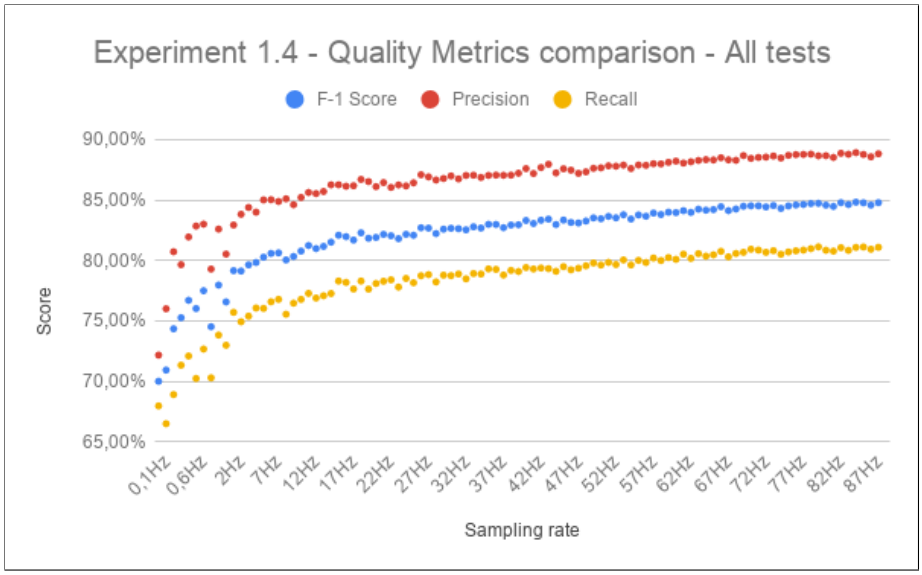

Figure 5.34: Experiment 1.4: Quality metrics comparison of the tests using set between $0.1Hz$ and $90Hz$. Note that the graph is not to scale, since the first ten sampling rates implement a logarithmic scale. *No. trees = 32; Train samples = (8,100,000 - 9,000); Test samples = (900,000 - 1,000)*

5.5 Experiment 2.1

The first experiment for the windowed implementations aims to provide proof of the chosen window size. This experiment tests twenty different setups, with window sizes between 50 and 1000 data instances each. This corresponds to windows between $0.5s$ and $10s$ worth of data. Each experiment uses the whole original dataset, and discards windows including more than one activity. Each subset is divided in two, 66% is used as a training set and 33% as the test set.

Results:

Figures 5.35 and 5.36 represent both the confusion matrix and the quality metrics of the system using $1.5s$ windows, the second best performer in the experiment. Figures 5.37 and 5.38, on the other hand, show the same information from the $6s$ windows test, the best performer. All the results can be compared in Figure 5.39, where the $6s$ window system has the best F_1 score. It is noticeable that both the $6s$ and the $9s$ systems have a contrasting performance compared to its neighbors. It can also be noted that the smaller the window sizes, the more constants the results are.

Conclusion:

The conclusion from this experiment is that window sizes between $0.5s$ and $4s$ have extremely constant results; lengthier windows, those over $4s$, appear to be more randomly distributed, with some systems, such as the 600 samples system ($6s$) outperforming every other test, but bordered by some of the worst performing systems. Shorter windows appear to discard less samples during generation, which leads to more data instances and results in better training. They also have more samples from the same amount of original data, since windows are shorter, meaning that the classifier is more likely to encounter infrequent classes, such as windows labeled as bending. Coincidentally, bending is a short duration action, which makes it be discarded more often in windows over $4s$, leading to it being increasingly rarer. It is decided, therefore, that future systems will be using $1.5s$ windows. This decision comes from the fact that this measurement appears to be amongst the most consistent set of tests, and still outperforms all other sub $4s$ implementations, with an F_1 score of 86.54%.

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	48	4	0	0	23	13	28	116	0,53%
	b'cycling'	1	1257	0	0	16	22	17	1313	5,95%
	b'lying'	0	1	1318	0	85	6	28	1438	6,52%
	b'running'	0	0	0	131	12	9	36	188	0,85%
	b'sitting'	3	7	16	6	12786	90	52	12960	58,76%
	b'standing'	4	15	13	1	116	2587	140	2876	13,04%
	b'walking'	7	15	23	5	143	144	2826	3163	14,34%
	TOTAL	63	1299	1370	143	13181	2871	3127	22054	100,00%
	PERCENTAGE	0,29%	5,89%	6,21%	0,65%	59,77%	13,02%	14,18%	100,00%	

Figure 5.35: Experiment 2.1: Confusion matrix of the 1.5*s* window system. *No. trees = 32; Train samples = 44,776; Test samples = 22,054*

							MACRO	
PRECISION	76,19%	96,77%	96,20%	91,61%	97,00%	90,11%	90,37%	91,18%	PRECISION
RECALL	41,38%	95,73%	91,66%	69,68%	98,66%	89,95%	89,35%	82,34%	RECALL
F-1 SCORE	53,63%	96,25%	93,87%	79,15%	97,82%	90,03%	89,86%	86,54%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	95,01%	ACCURACY

Figure 5.36: Experiment 2.1: Quality metrics of the 1.5*s* window system. *No. trees = 32; Train samples = 44,776; Test samples = 22,054*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	7	1	0	0	4	1	2	15	0,28%
	b'cycling'	1	298	0	0	9	5	4	317	5,98%
	b'lying'	0	1	315	0	31	2	15	364	6,86%
	b'running'	0	0	2	37	5	0	7	51	0,96%
	b'sitting'	0	3	3	0	3220	17	15	3258	61,43%
	b'standing'	0	5	3	1	45	566	14	634	11,95%
	b'walking'	0	2	5	1	51	17	589	665	12,54%
	TOTAL	8	310	328	39	3365	608	646	5304	100,00%
	PERCENTAGE	0,15%	5,84%	6,18%	0,74%	63,44%	11,46%	12,18%	100,00%	

Figure 5.37: Experiment 2.1: Confusion matrix of the 6*s* window system. *No. trees = 32; Train samples = 10,766; Test samples = 5,304*

							MACRO	
PRECISION	87,50%	96,13%	96,04%	94,87%	95,69%	93,09%	91,18%	93,50%	PRECISION
RECALL	46,67%	94,01%	86,54%	72,55%	98,83%	89,27%	88,57%	82,35%	RECALL
F-1 SCORE	60,87%	95,06%	91,04%	82,22%	97,24%	91,14%	89,86%	87,57%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	94,87%	ACCURACY

Figure 5.38: Experiment 2.1: Quality metrics of the 6*s* window system. *No. trees = 32; Train samples = 10,766; Test samples = 5,304*

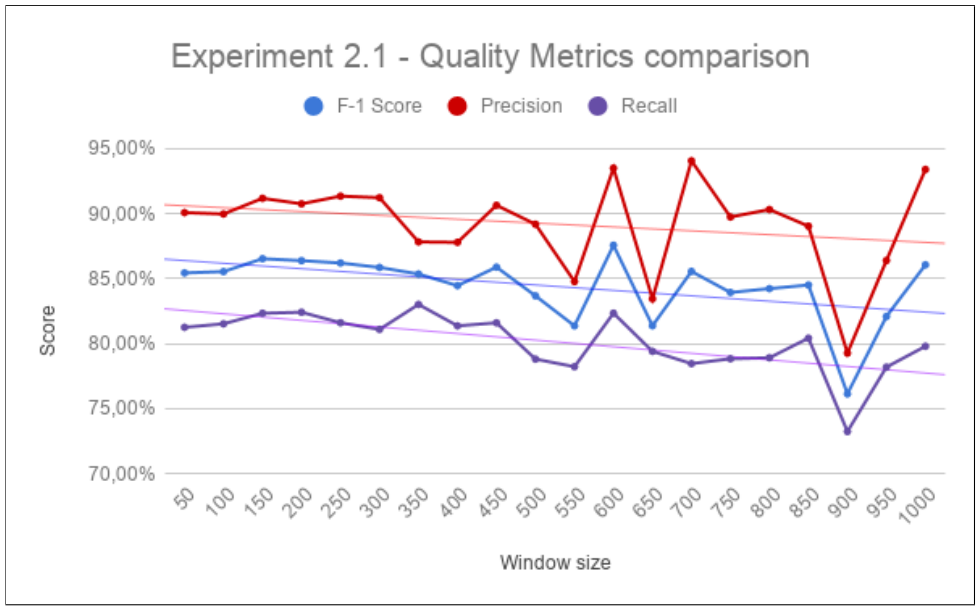

Figure 5.39: Experiment 2.1: Quality metrics comparison of all systems implemented for experiment 2.1. Systems implemented window sizes between 0.5s and 10s, which included between 50 and 1000 samples respectively. *No. trees = 32; Train samples = (135,541 - 6,282); Test samples = (66,760 - 3,095)*

5.6 Experiment 2.2

In experiment 2.2, the altered value is the sampling rate when generating windows. In the previous experiment, it was concluded that windows with 150 samples have the most constant results, which means 1.5s worth of accelerometer data. Hence, the 4Hz test within a 1.5s window will include 6 accelerometer readings, from which all features will be extracted. This test compares systems ranging between 2Hz and 100Hz. The 1Hz system was excluded since it would only have a single measurement, from which most of the features could not be extracted. Windows including half a value were rounded down. All tests include similar amount of data samples, but the difference between them is the density of the window.

Results:

Figure 5.40 shows all the quality metrics compared. A downward trend can be observed, as well as a lot of deviation from the tendency line. Several flat sections are noticeable, mainly around the 45Hz, the 60Hz, and between the 20Hz and 30Hz marks. Figures 5.41 and 5.42 are the confusion matrix and quality metrics

of the $44Hz$ example, which is one of the best performers in the experiment and is surrounded by very similar values. In contrast, Figures 5.43 and 5.44 showcase the $19Hz$ test, where the F_1 score is lower than in neighboring tests.

Conclusion:

This experiment has very inconsistent results, a high variance in the F_1 score of nearby systems is observed. Several clusters of flat performance exist, mainly between $40Hz$ - $50Hz$ and between $58Hz$ - $68Hz$. Consistency is of severe importance in scientific studies, and these two ranges outperform the rest in this regard. A HAR system using $1.5s$ windows sampled at $40Hz$ consumes 60% less energy than one sampled at $100Hz$, and the results are comparable. Therefore, the conclusion from this experiment is that lowering the sampling rate to somewhere in between $40Hz$ and $70Hz$ is beneficial for the system.

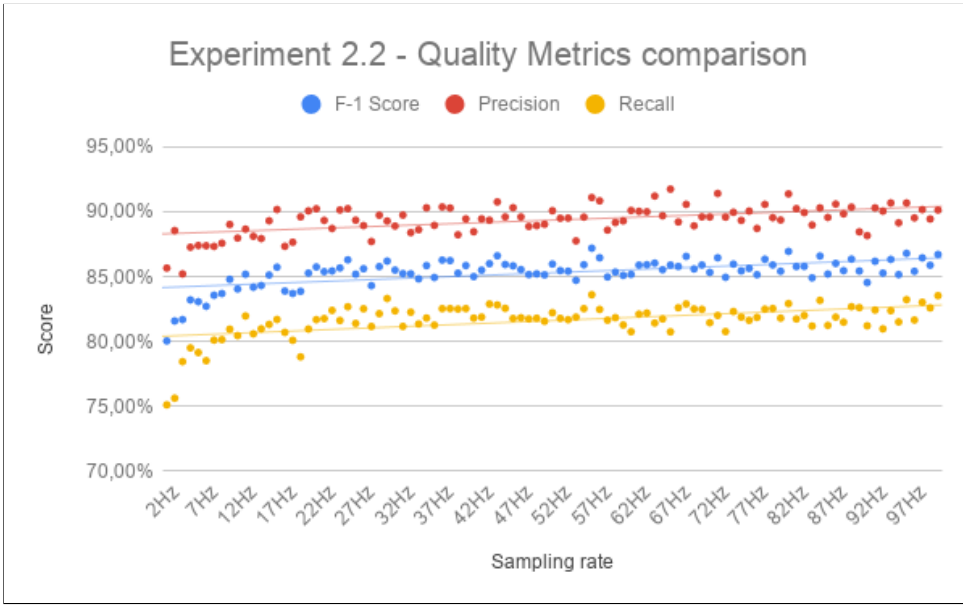

Figure 5.40: Experiment 2.2: Quality metrics comparison of the tests using windows captured with sampling rates between $2Hz$ and $100Hz$. *No. trees = 32; Train samples = 45,000; Test samples = 22,500*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	55	6	0	1	20	14	21	117	0,53%
	b'cycling'	2	1170	1	1	21	19	20	1234	5,60%
	b'lying'	0	4	1275	0	80	7	29	1395	6,33%
	b'running'	0	3	9	139	12	4	36	203	0,92%
	b'sitting'	4	18	12	4	12913	76	75	13102	59,41%
	b'standing'	6	17	9	0	105	2638	141	2916	13,22%
	b'walking'	7	17	16	5	134	159	2749	3087	14,00%
	TOTAL	74	1235	1322	150	13285	2917	3071	22054	100,00%
	PERCENTAGE	0,34%	5,60%	5,99%	0,68%	60,24%	13,23%	13,92%	100,00%	

Figure 5.41: Experiment 2.2: Confusion matrix of the $44Hz$ system. *No. trees = 32; Train samples = 44,774; Test samples = 22,054*

							MACRO	
PRECISION	74,32%	94,74%	96,44%	92,67%	97,20%	90,44%	89,51%	90,76%	PRECISION
RECALL	47,01%	94,81%	91,40%	68,47%	98,56%	90,47%	89,05%	82,82%	RECALL
F-1 SCORE	57,59%	94,78%	93,85%	78,75%	97,87%	90,45%	89,28%	86,61%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	94,94%	ACCURACY

Figure 5.42: Experiment 2.2: Quality metrics of the $44Hz$ system. *No. trees = 32; Train samples = 44,774; Test samples = 22,054*

		PREDICTED								
		b'bending'	b'cycling'	b'lying'	b'running'	b'sitting'	b'standing'	b'walking'	TOTAL	PERCENTAGE
ACTUAL	b'bending'	38	14	0	0	24	15	27	118	0,53%
	b'cycling'	1	1195	0	0	21	36	20	1273	5,67%
	b'lying'	0	4	1322	4	97	8	34	1469	6,54%
	b'running'	0	2	7	126	18	9	49	211	0,94%
	b'sitting'	1	17	18	4	13075	99	108	13322	59,34%
	b'standing'	1	13	7	2	111	2650	168	2952	13,15%
	b'walking'	10	11	18	6	150	175	2736	3106	13,83%
	TOTAL	51	1256	1372	142	13496	2992	3142	22451	100,00%
	PERCENTAGE	0,23%	5,59%	6,11%	0,63%	60,11%	13,33%	13,99%	100,00%	

Figure 5.43: Experiment 2.2: Confusion matrix of the $19Hz$ system. *No. trees = 32; Train samples = 45,581; Test samples = 22,451*

							MACRO	
PRECISION	74,51%	95,14%	96,36%	88,73%	96,88%	88,57%	87,08%	89,61%	PRECISION
RECALL	32,20%	93,87%	89,99%	59,72%	98,15%	89,77%	88,09%	78,83%	RECALL
F-1 SCORE	44,97%	94,50%	93,07%	71,39%	97,51%	89,17%	87,58%	83,87%	F-1 SCORE
ACCURACY	-	-	-	-	-	-	-	94,17%	ACCURACY

Figure 5.44: Experiment 2.2: Quality metrics of the $19Hz$ system. *No. trees = 32; Train samples = 45,581; Test samples = 22,451*

5.7 Experiment 2.3

Experiment 2.3 is the main experiment of this master's thesis. A total of 15,979 distinct scenarios were tested. These three variables were altered: Window density, window distance, and window size. Window density ranged between $90Hz$ and $2Hz$, with $1Hz$ steps; window size ranged between $1.5s$ and $10.0s$, with $0.5s$ steps; window distance ranged between one and 10 full windows between consecutive windows, meaning that a $3s$ window was tested with 3, 6, 9, 12, 15, 18, 21, 24, 27, and 30 seconds window distance. The test consisted of a RF classifier with 66% - 33% train-test division.

Results:

The results for this experiment include more data points than previous ones and are, therefore, harder to visualize. Figure 5.45 is a scatter plot representing all the F_1 score values plotted against the amount of samples used both for training and testing in that test. Note that this graph uses a logarithmic scale on the horizontal axis. The tendency function is shown in Figure 5.46, which displays a downward trend in the quality metrics linked to lower amounts of samples. Three sections are identifiable in Figure 5.45: Between 0 and 10,000 samples a disperse area is noticeable where results range between the 60% and 90% marks; between 10,000 and 1,000,000 samples the area contains most of the test results, and shows a packed score around the 80%; the tests with more than 1,000,000 samples show a consistent better performance, around the 85% mark. The ten results with the best performance are isolated in Figure 5.47. Any one of these ten tests has a better performance than any other result with lower amount of used samples, and together they form, therefore, the Pareto front.

Conclusion:

Similar to the results in Experiment 1.4, the systems trained using fewer samples have lower dependability. Albeit some systems have better performance than systems using more samples, the trend is that most systems on the right hand side of Figure 5.45 have better results than the average on the left side of the graph. The best performers, though, are systems trained with somewhere in between 9,000 and 90,000 samples, and all achieved over 92% F_1 score. These systems all use lengthier windows than what was seen in previous experiments, over $8s$, and the window distance is also longer than expected, since all of them have over a minute between consecutive windows.

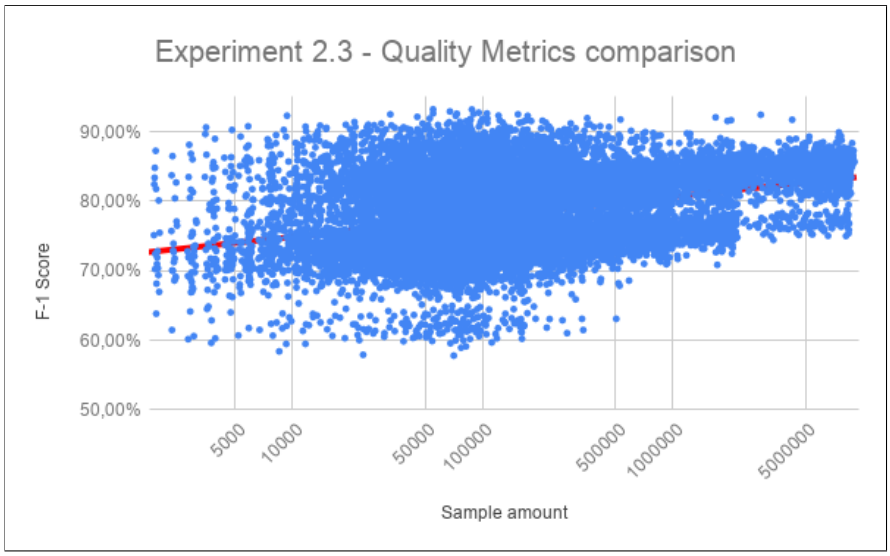

Figure 5.45: Experiment 2.3: F_1 score comparison of all tests. *No. trees = 32; Amount of samples used for the test = (9,022,050 - 1,870)*

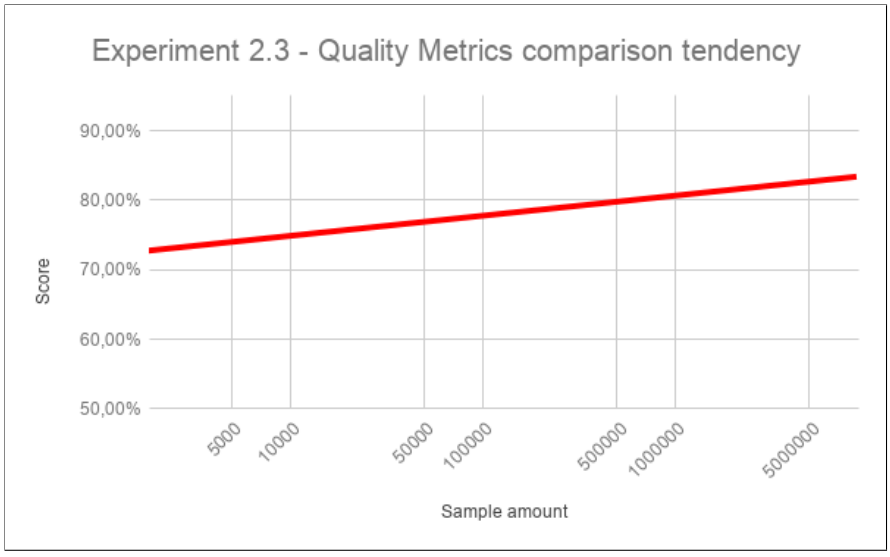

Figure 5.46: Experiment 2.3: Tendency of the F_1 score according to the amount of samples used for training and testing. *No. trees = 32; Amount of samples used for the test = (9,022,050 - 1,870)*

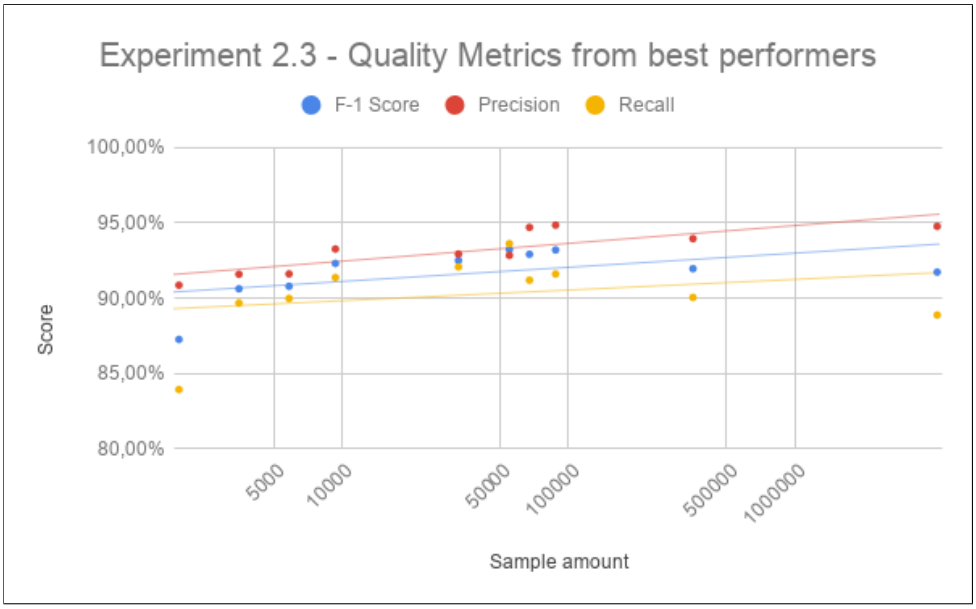

Figure 5.47: Experiment 2.3: Precision, recall and F_1 score of the best performer in each group. *No. trees = 32; Amount of samples used for the test = (4,252,095 - 1,910)*

5.8 Experiment 2.4

The last experiment consists of using the top performers from Experiment 2.3 as train tests and utilizing all other sets as test sets. The test sets have the same variables as in the previous experiments, those being: Window density, ranged between $90Hz$ and $2Hz$, with $1Hz$ steps; window size, ranged between $1.5s$ and $10.0s$, with $0.5s$ steps; and window distance, ranged between one and 10 full windows between consecutive windows. Table 5.1 displays the characteristics of the ten training sets used in this experiment.

Results:

Figure 5.48 represents the F_1 score comparison of all the tested systems. The graph displays several vertical lines of results with different scores, symbolizing each step of a lower amount of samples. It can be noticed that the datapoints follow the same pattern as in Figure 5.46 from experiment 2.3. It can also be observed that some results have extremely high values, with over 99% F_1 score.

Total samples	Sampling rate	Window size	Window distance	F_1 score
1910	$2Hz$	$7s$	$70s$	87.26%
3515	$3Hz$	$8s$	$72s$	90.63%
5850	$4Hz$	$10s$	$80s$	90.79%
9379	$10Hz$	$10s$	$100s$	92.31%
32738	$28Hz$	$8.5s$	$85.5s$	92.51%
54945	$37Hz$	$8s$	$72s$	93.23%
67390	$46Hz$	$10s$	$80s$	92.92%
87900	$60Hz$	$10s$	$90s$	93.20%
355215	$60Hz$	$9.5s$	$38s$	91.96%
4252095	$45Hz$	$9s$	$9s$	91.74%

Table 5.1: Top ten performers and their specifications from experiment 2.3

Conclusion:

It is nearly impossible for a classifier to attain 100% precision, but that result can be seen in this experiment. The cause of this is that the training and testing sets share some of the data instances, which is one of the rules in supervised learning that should not be broken, as was explained in Section 2.2. As stated in the supervised learning subsection: *The test set is composed of examples distinct from those in the training set.* Due to the way different sets are generated in this thesis, from the same original dataset, it is likely that different subsets share some datapoints. This allows the classifier to train specifically for the test and does not reflect the real performance of the system. This experiment was not conducted using a reliable method, and does not fulfill the objective proposed in the methodology section. It is, however, unfeasible to generate the datasets required to run it manually, as it takes between 20 and 30 seconds to generate each subset, which equates to 150 hours for all the 16,000 sets generated for experiment 2.3.

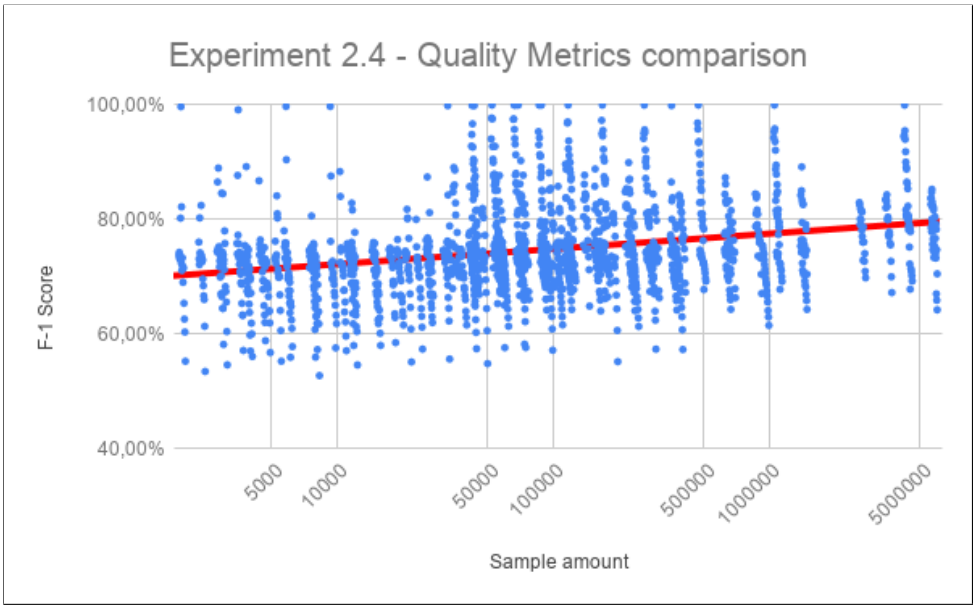

Figure 5.48: Experiment 2.4: F_1 score comparison of all tests. *No. trees = 32; Amount of samples used for the test = (6,014,610 - 1,870)*

Chapter 6
Discussion

This section serves as an examination of the experiments run for this thesis as a whole and an analysis of their outcomes. It is divided in three sections: the first part is specific to the beginning round of experiments, including experiments 1.1, 1.2, 1.3, and 1.4; the second part includes the results of all the remaining experiments, those including windowing techniques; the third section serves as an examination of the procedure carried out for this master's thesis and as a deliberation on the resulting outcome.

6.1 Windowless experiments

Human Activity Recognition systems include a segmentation phase, according to the Activity Recognition Chain described in Figure 2.1. During the segmentation phase windows are created and labeled to be used in the classification phase. Throughout experiments 1.1, 1.2, 1.3, and 1.4, though, no windows were used. This is not the common procedure in HAR, but the potential usefulness of a classifier that is precise enough to be competitive and uses only one measurement per label is warrant enough to experiment with windowless systems. This potential system could use as low as one measurement to identify the ongoing activity, but only if the classifier is reliable enough and has been extensively trained.

Intention:

Previous studies attain better performance when using windows, which means that the systems with the best performance all use a windowed implementation that simplifies the windows through features. Feeding the raw data to the classifier complicates the classification task for the random forest, making the training process lengthier, as well as resulting in a forest containing trees with more depth and more nodes. Complicating the training task yields worse classifiers, which make poor use of the training data and need more data instances for similar performance. On the one hand, the conclusion from these statements is that the best classifiers, as has

been seen throughout the literature, use windows. On the other hand, none of these facts focus on how a windowed classifier reacts to lower data amounts in comparison with non-windowed implementation. The intention of the first four experiments was, therefore, exploring this exact scenario. The hypothesis was that windowless systems would react differently to lower amounts of training data compared to similar systems using windows.

Results:

The aim of **Experiment 1.1** was making a distinction between sampling rate and amount of samples. The sampling rate exclusively means the amount of times per second the sensor captures data. Naturally, and in the same amount of time, two sensors with the same sampling rate capture the same amount of data, and if one of those has double the sampling rate it would capture twice as much data instances as the other sensor. With variable time frames, however, sampling rate does not give any information, since a sensor with a lower sampling rate might be running longer and record more data instances. Experiment 1.1 proves this statement by training and testing a classifier with the same total amount of data instances, but which was captured at different sampling rates. In order to have the same sample quality, the original set was shuffled before down-sampling. Quality refers to the composition of the set, meaning that the data contains instances of each class roughly in the same percentages as were shown in Table 4.2. The results showcased in Figure 5.5 clearly display a flat trend in the quality metrics, meaning that a change in the sampling rate has no effect if the amount of training and testing data is the same in the end.

Experiment 1.2 was aimed at defining the effects that varying the size of the test set has in the performance of a given system. With this, the viability of having an extensively trained system and using only a handful of samples to test a given moment could be determined. The results were rather interesting, since some of the best performers had very few samples. The $0.2Hz$ system had the best F_1 score, of 83%. As counterintuitive as that may seem, the cause is that smaller sample sizes lead to potentially random results, given that the deciding instances might only appear a relatively low amount of times. In the dataset used in this thesis, the *bending* category was the worst performer, so having only one instance of it and labeling it correctly would not reflect the actual performance of the classifier, which is what can be seen in Figure 5.15 and 5.14, where *bending* has metrics outperforming those seen in the $100Hz$ case, in Figure 5.8. With this results, it can be extrapolated that insufficient amount of test samples lead to random results, that can be falsely over-performing systems with more samples, which are more consistent.

In order to specify the required amount of data samples to run a reliable test, in **Experiment 1.3**, a single subset was used to test several training sets of different sizes. The expectation was that a specific amount of minimum samples would be required to have a successful test, but the actual outcome advocates that using a percentage of the training set as test always has better performance. As was seen in Figure 5.26, fixed size test sets have worse performance than relative sized ones. The conclusion from this experiment is the following relationship between the classifier's worst case performance and the percentage of samples that case has, and how those affect the minimum sample percentage for a test case. In the dataset used in this thesis, the most complicated class to classify is the *bending* class. That category represents 0.58% of the total data instances, as seen in Table 4.2. Pairing the 0.58% and the performance of the classifier for that label seen in Figure 5.3 (51.92% F_1 score for the *bending* class) it is concluded that 30.11% ($30.11 == 0.58 \times 51.92$) of the total amount of data instances should be used for testing in order to obtain completely reliable results. As seen throughout the experiment, anything lower than 10% is not stable enough to build a system. For studies where the maximum performance is needed, 30% is the perfect amount, being 10% the minimum for a trustworthy system that does not depend on randomness.

After determining the minimum and the optimum test sizes, **Experiment 1.4** was run to determine the minimum and optimum train sizes. It is observable in Figure 5.34 that datasets sampled at less than $10Hz$ (containing 900,000 training samples) underperform when compared to systems using more samples for training. Moreover, when training with less than 270,000 samples (at the $3Hz$ mark) results appear to be inconsistent, which makes those systems unusable for a scientific scenario. Those results, similarly to the conclusion from previous experiments, depend enough on the random factor of underperforming classes appearing a specific amount of times that are not considered to be suitable as a design choice for a Human Activity Recognition system.

Resolution:

After grouping the results of all the experiments run for this section, as a minimum, HAR systems not implementing windows and implementing a Random Forest classifier must utilize a minimum of 500,000 samples, which should be further divided into two subsets, one containing 90% and the other 10% of the total samples. The 90% subset should be used for training, and the remaining set, for testing. The obtained data samples do not need to be gathered at a specific sampling rate, but must be varied enough to represent the classes wishing to be identified. Using these parameters, this study attained an F_1 score of 79.84%. In continuous studies this amount of data instances correspond to a sampling rate of $5Hz$.

Systems with access to more data would benefit from using a minimum of 1,900,000 total samples (or $19Hz$), and training the RF with 70% of those samples, meaning that 30% would be left for testing it. Within this master's thesis, the proposed system attained an F_1 score of 82.29%, which is around 2.5% better performance than the reduced system previously proposed. This system, however, would use as much as 280% more energy than the reduced proposal, making it less efficient.

6.2 Windowed experiments

As seen throughout the studied literature, current HAR systems group consecutive data instances in windows which are then simplified through features that describe them. This approach makes the classification task easier and the system has better overall performance. Since the usage of windowing techniques is common in modern systems, exploring and experimenting with those same methods is mandatory in order to make this research relevant. Experiments 2.1, 2.2, 2.3, and 2.4 make use of windowing techniques and explore the effects that lowering the amount of samples used for training and testing has on the performance of the system.

Intention:

By altering the amount of samples a sensor needs to gather during a time frame, the sampling rate, better use of battery-life can be reached. The intention of these experiments was reaching a conclusion regarding how a lower amount of samples affects the performance of a HAR classifier, which would allow the plotting of its efficiency. In order to achieve this, the proposed method is based on the comparison of multiple implementations of the same system, with the input data coming from the same dataset and describing the same activities, but altering the amount of data instances used from within that original dataset. In a windowed implementation, in order to alter the amount of data samples used, two possibilities exist: either the density of the window is changed, meaning the amount of samples per second within that window; or the distance between consecutive windows is adjusted. In addition to these two, other parameters, such as the window size, might have an effect on the results. The three variables that form the experiments are, therefore, window size, window density, and window distance.

Results:

Experiment 2.1 focuses on the window size, the aim is that of establishing a benchmark on how the window size affects the performance of a classifier, for future reference in the following experiments. Human activities are linked to physical actions and those depend on time, so it is natural that the amount of seconds a window encompasses affects the result of the classifier. In experiment 2.1, window sizes of up

to ten seconds were tested. As seen in Figure 5.39, the best performers are the $6s$ and the $10s$ window, but the smaller window sizes have more constant results, which do not deviate from their neighbors as much. Confusing classes for the classifier, such as the *bending* action, appear to be short in duration, which makes larger windows more likely to discard them. This has two effects: on the one hand, it lowers the amount of instances of those classes, which leads to worse training. On the other hand, it cleans the dataset of transitional activities, those being the instances of an action that are labeled as such, but actually include the transition from a previous action to the labeled one. Transitional activities complicate the classification task and ridding the dataset of those results in better performance. A combination of these two facts is what makes certain implementations, such as the $10s$ system, outperform the rest.

In **experiment 2.2**, the altered value was the window density, referring to the sampling rate of the accelerometer from which the data was generated. Denser windows require more data samples, hence more energy. A system can be considered more efficient if windows used in it are less dense but results are similar. Altering the density, as opposed to the amount of windows, does not affect the amount of training samples the classifier has access to as long as the rest of parameters are constant. Experiment 2.2 compared $1.5s$ windows with different densities, ranging between $2Hz$ and $100Hz$. The results in Figure 5.40 show a downward trend with lower densities, but the rate of decay is quite flat, specially after the $20Hz$ mark. This can be attributed to the statement by Karantonis et al., *All measured body movements are contained within frequency components below 20 Hz [KNM+06].* From combining experiments 2.1 and 2.2, it can be concluded that a system with $10s$ windows sampled at $20Hz$ (so containing 200 data instances) does not need to sacrifice any performance and offer results comparable to state-of-the-art systems.

Experiment 2.3 was the main experiment within this master's thesis, a total of 16,000 different datasets were created and used to train and test a classifier. All the results are plotted in Figure 5.45, where a downward trend linking lower sampling rates to lower performance is observable. This trend, however, does not mean that certain systems cannot be more efficient than others, as is noticeable by some of the systems that used less than 5,000 samples and attained over 90% F_1 score. The takeaways from this experiment are the common characteristics of these systems, so that future studies can use those traits and use them. As seen in **Experiment 2.4**, in Table 5.1, the top performing systems tend to have a larger-than-usual window size, between 7 and 10 seconds, with an average window size of 9s. The best performer was the system using windows encompassing $8s$ worth of measurements at $37Hz$, with a window distance of $72s$. This system attained an F_1 score of 93,23%, and used a total of 32,728 samples both for training and testing a system from 28h of labeled data. Compared to the systems resulting from experiment 1.4, the F_1 score is 10.93% higher, and the energy usage over 58 times lower.

Resolution:

From all the experiments between 2.1 and 2.4, the following facts can be extracted. Longer window sizes, between 7 and 10 seconds, have been proven to perform better than shorter counterparts, between 0.5 and 4 seconds. Shorter windows, however, appear to be more constant in their results, as seen in Figure 5.39 from experiment 2.1, which allows the study implementing the system to not worry about their window size and its performance. Longer window sizes, albeit performing better in the best-case scenario, have a higher standard deviation, which results in the necessity for more fine tuning of the system. Longer window sizes are also more resilient to working with less data, as established in Table 5.1 from experiment 2.4, where all the top performers have larger window sizes and use less overall data instances.

Given the slow nature of human physical activity, where actions last periods of time of dozens of seconds, window distances of over $60s$ prove to be more battery efficient when comparing the system's performance. Meaning that by skipping $60s$ of measurement between windows most of the energy expenditure can be lowered and the system still yields highly accurate results. These results are extracted from Figure 5.47, where the average distance between consecutive windows is around the $70s$ mark, being all the top performers above $72s$.

Reducing the sampling rate from within a window does not affect the amount of windows the trainer will work with, but instead their quality, which comes from the amount of samples they contain. The features generated from those samples are affected by the window density, and features such as the zero cross rate (the amount of times the population goes from positive to negative or vice versa) do not hold as much information when working with sparser windows, since they are quantifiable values that depend on the size of the population. The Pareto front resulting from experiment 2.3 contains window densities (their sampling rates) of 2, 3, 4, and 10 Hz, which are considered low when comparing them to studied HAR systems; it also contains higher rates, of 28, 37, 46, and 60 Hz, and those have an average of 2.3% better performance, but also use 9 times more samples, on average.

6.3 Outcome

Future studies that need battery efficient systems would benefit from implementing classifiers in the Pareto front of Figure 5.45, which are all summarized in Table 6.1 and plotted in Figure 6.1:

The first system has an overall performance of 87.26%, 5.97% lower than the top performer, but is over 27 times more battery efficient, using a total amount of samples for training and testing of 1,910.

The second system has an overall performance of 90.63%, 2.60% lower than the top performer, but is over 15 times more battery efficient, using a total amount of samples for training and testing of 3,515.

The third system has an overall performance of 90.79%, 2.44% lower than the top performer, but is over 9 times more battery efficient, using a total amount of samples for training and testing of 5,850.

The fourth system has an overall performance of 92.31%, 0.92% lower than the top performer, but is over 5 times more battery efficient, using a total amount of samples for training and testing of 9,379.

The fifth system has an overall performance of 92.51%, 0.72% lower than the top performer, but is over 67% more battery efficient, using a total amount of samples for training and testing of 32,738.

The top performing system has an overall performance of 93.23%, and uses a total amount of samples for training and testing of 54,945. This system does not use extremely low amounts of samples, and performs better than any other system seen throughout this thesis. It implements window sizes of $8s$, distanced by $72s$ between measurements, and those windows are composed of 296 samples, corresponding to a $37Hz$ sampling rate.

By combining window density, window distance, and window size as parameters for the generation of data sets, lower sampling rates have been proven to be overcome. From the proposed systems, the implementation using 3,515 samples has an F_1 score of over 90%, and its features simplify 28h of data gathered at $100Hz$ in less than 2,855 times the previous energy expenditure.

Total samples	Sampling rate	Window size	Window distance	F_1 score
1910	$2Hz$	$7s$	$70s$	87.26%
3515	$3Hz$	$8s$	$72s$	90.63%
5850	$4Hz$	$10s$	$80s$	90.79%
9379	$10Hz$	$10s$	$100s$	92.31%
32738	$28Hz$	$8.5s$	$85.5s$	92.51%
54945	$37Hz$	$8s$	$72s$	93.23%

Table 6.1: Classifiers in the resulting Pareto front.

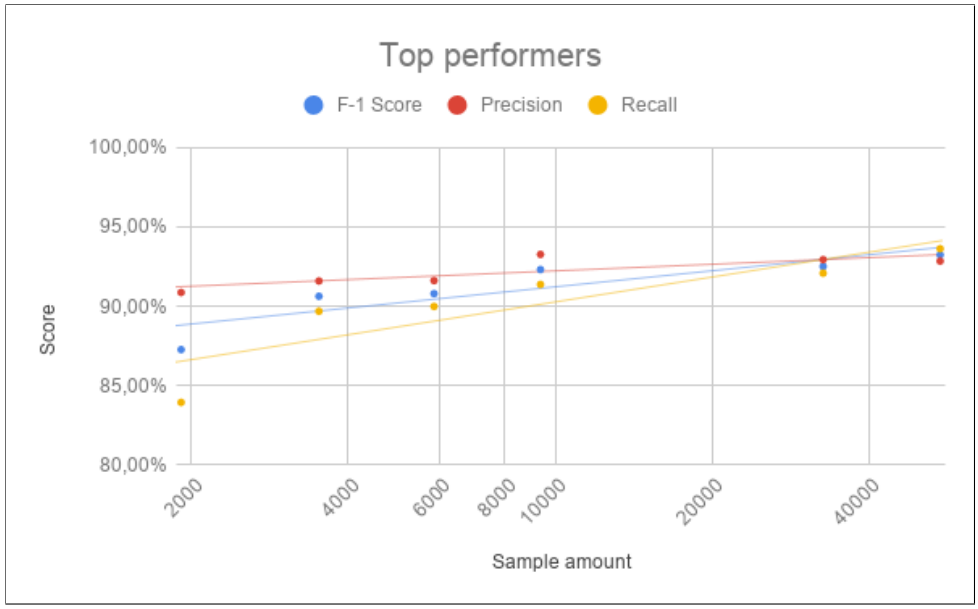

Figure 6.1: Quality metrics comparison of the top performing systems.

Chapter 7
Conclusion

After the discussion of the experiments run for this master's thesis, the work is considered to be concluded. The research goals have been successfully achieved, and an answer has been given to all the questions proposed at the beginning of this project. The result is the implementation of six systems that achieve F_1 scores between 87.26% and 93.26%, and use sample sets with as low as 1,910 data instances. These systems all label human activity by using a Random Forest classifier composed of 32 trees, and use 66% of the given set for training and the other 33% for testing. The data used was gathered by two accelerometers, one on the front-right thigh and another on the upper-left back.

Over 16,000 different tests were run for this master's thesis, in order to find the best parameter combination that would be more resilient to using lower amounts of samples. These tests included windowed and non-windowed implementations, as well as several variables to be combined within those. The non-windowed tests varied the amount of samples, train-test division, and sampling rate. The windowed tests varied the train-test division, window distance, window size, and window density.

7.1 Contribution

The objective of this master's thesis was the exploration of the resilience of HAR systems to lower sampling rates. By implementing several systems with different characteristics the objective is considered to be fulfilled. Different random forests have been implemented, by using variations of the original data set, and their results compared to one another. This completes the first goal of this thesis, which was training different RFs with variations of the same data set using lower sampling rates. The answer to the first research question, which was: *What is the minimum amount of training data required to have a dependable Random Forest?* is the first proposed system. The classifier uses 1,910 total samples for training and testing, and has an F_1 score of over 87%. The second question was about the effect that windowing

techniques have on the dependence of RFs on the sampling rate. The answer to the second question was exemplified during the discussion, where all the systems using windows performed better than the best system not using them, and most of them needed fewer data instances for their results.

By combining all the implemented systems, a Pareto front has been established. Six systems form it and are the most performant systems from all the tested examples throughout this thesis. The establishment of the Pareto front fulfills the second goal established for this project. Answering question number three, the more vulnerable systems to lower sampling rates are non-windowed implementations, which were proven to be ineffective when trained with less than 500,000 samples. After comparing all the systems, Random forests with 32 trees which use a sample set created at $2Hz$ and $7s$ windows are the most resilient systems, which is the answer to question number four.

7.2 Future work

There are many clear paths that need to be traversed that this thesis could not cover, among those some are features that did not fit the scope of the project, and some are similar but alternative situations that are potentially interesting:

First and foremost, an experiment similar to experiment 2.4 should be properly conducted, where the amount of necessary samples for testing a classifier is determined. The generation of new subsets was a task too computationally expensive to be performed many times, and experiment 2.3 took over three days to prepare. Even if it could not be conducted, the necessity of obtaining results such as those sought in experiment 2.4 could warrant a new line of work.

After exploring the random forest classifier, many other machine learning types remain. These include Support Vector Machines, k-Nearest Neighbors, Random Forests, and Artificial Neural Networks, among many others commonly seen in HAR. The same type of study could be conducted for any single one of those systems, or use them all together as a new variable.

The initial proposal for this master's thesis included the exploration of adaptive sampling, where the time between consecutive samples would vary according to the last classified label. It was not possible due to the magnitude of the project to implement smart sampling systems, but the proposal has a lot of potential and could be continued in that direction.

This project serves as a proof of concept for what it succeeds in, but actual validation outside of the lab and the usage of more realistic setups is mandatory for it to be applied to consumer applications.

As a natural continuation from this project, results under $1Hz$ could be experimented with. Given the tested window sizes, of a maximum of $10s$, some windows would end up empty if rates of $0.01Hz$ were used, but if that were the objective the minimum window size could be two minutes. This scenario is a possible extension to this project.

The focus has been always on the same dataset, composed of data from two accelerometers and created for another purpose (early stroke detection). Not only a specific dataset could be created for the project, where sets at lower rates are generated for each test, but results could also be tested with other already existing datasets, such as the HAR smartphone dataset[1]. New labels could also be used, or some removed, as it would allow the removal of confusing labels such as the *bending* label, which might allow even lower amounts of samples to be used.

Other types of pattern detection, not only those in human activity, but maybe also noise, or images, are a potential path to follow. Noise detection is an important topic for Internet of Things, and its importance in the future might depend on studies like this one applied to it. Performing the same type of study in those environments is another of the paths that could be followed.

[1] https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

References

[BBS14] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.

[BKV+97] Carlijn VC Bouten, Karel TM Koekkoek, Maarten Verduin, Rens Kodde, and Jan D Janssen. A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE transactions on biomedical engineering*, 44(3):136–147, 1997.

[EHM16] Obiora Sam Ezeora, Jana Heckenbergerova, and Petr Musilek. A new adaptive sampling method for energy-efficient measurement of environmental parameters. In *Environment and Electrical Engineering (EEEIC), 2016 IEEE 16th International Conference on*, pages 1–6. IEEE, 2016.

[ESCD+18] Rayane El Sibai, Yousra Chabchoub, Jacques Demerjian, Raja Chiky, and Kablan Barbar. A performance evaluation of data streams sampling algorithms over a sliding window. In *Communications Conference (MENACOMM), IEEE Middle East and North Africa*, pages 1–6. IEEE, 2018.

[FKL17] Liang Feng, Pranvera Kortoçi, and Yong Liu. A multi-tier data reduction mechanism for iot sensors. In *Proceedings of the Seventh International Conference on the Internet of Things*, page 6. ACM, 2017.

[HNK09] Yuya Hanai, Jun Nishimura, and Tadahiro Kuroda. Haar-like filtering for human activity recognition using 3d accelerometer. In *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th*, pages 675–678. IEEE, 2009.

[HT16] Hans-Olav Hessen and Astrid Johnsen Tessem. Human activity recognition with two body-worn accelerometer sensors. Master's thesis, NTNU, 2016.

[KC14] Narayanan C Krishnan and Diane J Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014.

[KLG11] Simon Kozina, Mitja Lustrek, and Matjaz Gams. Dynamic signal segmentation for activity recognition. In *Proceedings of International Joint Conference on Artificial Intelligence, Barcelona, Spain*, volume 1622, page 1522, 2011.

[KNM+06] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, 10(1):156–167, 2006.

[KP08] Narayanan C Krishnan and Sethuraman Panchanathan. Analysis of low resolution accelerometer data for continuous human activity recognition. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 3337–3340. IEEE, 2008.

[KSL13] Adil Khan, Muhammad Siddiqi, and Seok-Won Lee. Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones. *Sensors*, 13(10):13099–13122, 2013.

[KTKL14] Adil Mehmood Khan, Ali Tufail, Asad Masood Khattak, and Teemu H Laine. Activity recognition on smartphones via sensor-fusion and kda-based svms. *International Journal of Distributed Sensor Networks*, 10(5):503291, 2014.

[LKK11] Myong-Woo Lee, Adil Mehmood Khan, and Tae-Seong Kim. A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation. *Personal and Ubiquitous Computing*, 15(8):887–898, 2011.

[LL13] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209, 2013.

[LYC17] Song-Mi Lee, Sang Min Yoon, and Heeryon Cho. Human activity recognition from accelerometer data using convolutional neural network. In *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, pages 131–134. IEEE, 2017.

[MMH17] Yasser Mohammad, Kazunori Matsumoto, and Keiichiro Hoashi. A dataset for activity recognition in an unmodified kitchen using smart-watch accelerometers. In *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia*, pages 63–68. ACM, 2017.

[Ols01] Robert T Olszewski. Generalized feature extraction for structural pattern recognition in time-series data. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2001.

[OPB12] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How many trees in a random forest? In *International workshop on machine learning and data mining in pattern recognition*, pages 154–168. Springer, 2012.

[PFN06] Susanna Pirttikangas, Kaori Fujinami, and Tatsuo Nakajima. Feature selection and activity recognition from wearable sensors. In *International symposium on ubiquitious computing systems*, pages 516–527. Springer, 2006.

[PGKH08] Stephen J Preece, John Yannis Goulermas, Laurence PJ Kenney, and David Howard. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*, 56(3):871–879, 2008.

[PKA+17] Janne Paavilainen, Hannu Korhonen, Kati Alha, Jaakko Stenros, Elina Koskinen, and Frans Mayra. The pokémon go experience: A location-based augmented reality mobile game goes mainstream. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 2493–2498. ACM, 2017.

[PNW12] Orasa Patsadu, Chakarida Nukoolkit, and Bunthit Watanapa. Human gesture recognition using kinect camera. In *2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE)*, pages 28–32. IEEE, 2012.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[Rin12] Wolfgang Rindler. *Essential relativity: special, general, and cosmological*. Springer Science & Business Media, 2012.

[RN16] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[RZS12] Lingmei Ren, Quan Zhang, and Weisong Shi. Low-power fall detection in home-based environments. In *Proceedings of the 2nd ACM international workshop on Pervasive Wireless Healthcare*, pages 39–44. ACM, 2012.

[Sha98] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IEEE*, 86(2):447–457, 1998.

[Tei01] Jürgen Teich. Pareto-front exploration with uncertain objectives. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 314–328. Springer, 2001.

[Våg17] Eirik Vågeskar. Activity recognition for stroke patients. Master's thesis, NTNU, 2017.

[VBD+12] Jérémy Vanhelst, Laurent Béghin, Alain Duhamel, Patrick Bergman, Michael Sjöström, and Frédéric Gottrand. Comparison of uniaxial and triaxial accelerometry in the assessment of physical activity among adolescents under free-living conditions: the helena study. *BMC medical research methodology*, 12(1):26, 2012.

[WWF11] Feng Wang, Meiling Wang, and Nan Feng. Research on classification of human daily activities based on a single tri-axial accelerometer. In *Complexity and Data Mining (IWCDM), 2011 First International Workshop on*, pages 121–124. IEEE, 2011.

[Zha10] Neil Zhao. Full-featured pedometer design realized with 3-axis digital accelerometer. *Analog Dialogue*, 44(06):1–5, 2010.