NTNU
Norwegian University of
Science and Technology

Activity Recognition for Stroke Patients

Eirik Vågeskar

Master of Science in Computer Science
Submission date: July 2017
Supervisor: Helge Langseth, IDI
Co-supervisor: Kerstin Bach, IDI

Norwegian University of Science and Technology
Department of Computer Science

Summary

Stroke is a disruption in the blood flow to the brain which may lead to a death of brain cells. More than 12 000 Norwegians experience a stroke each year. Survivors often suffer lasting movement disabilities, which affect their ability to stay active and live independently. Knowledge about the time a patient spends performing certain activities in their daily life, as well the frequency and intensity with which they are performed, is useful for rehabilitation. Physical therapists can use this knowledge to adapt treatment plans to a patient's needs and abilities. Reports from larger groups of patients can be used to study the effects of new treatments.

Self-reporting and observation by a professional are the most common ways to acquire knowledge about how often and for how long a person performs different activities. These methods often lead to inaccurate results due to subjective errors and limited time respectively. Human activity recognition (HAR) systems are an alternative to these methods which is not prone to the same errors. Using machine learning techniques, these systems can recognize what activity a person performed at a given time. Measurements of the person's movements, made by one or more sensors worn on the body, are given to the system for it to perform its task. To learn how to perform this recognition, a HAR system must be trained on previous examples of such sensor recordings which have been labeled with what activity went on during each example. Because stroke patients move differently from healthy subjects, systems trained on examples from healthy subjects are inaccurate when recognizing the movements of stroke patients.

The goal of this thesis has been to create a HAR system which performs accurate activity recognition for stroke patients. This work has relied on a new data set collected from stroke patients. Fifteen stroke patients participated in the collection, wearing five three-axis accelerometer sensors, located on both wrists and thighs and the lower back. Hour-long recordings of the subjects were conducted in a laboratory, each subsequently labeled with the activities that the subject performed.

Experiments have involved using different combinations of accelerometers, classification techniques, training data sets, and amounts of training data, resulting in a classifier based on a random forests ensemble method. The final system is able to recognize the activities of stroke patients with 93.4% accuracy using one accelerometer on the lower back and one on the thigh. Using additional accelerometers on the thigh and wrist, the system achieves 94.6% accuracy. One of the final experiments shows that the system can be trained on a data set with examples from both healthy subjects and stroke subjects and still be equally accurate in recognizing activities for the subjects in both groups as classifiers targeting each group. This opens up the possibility of making HAR systems that are not tailored to specific patient groups, recognizing movements from both healthy subjects and subjects with different disabilities with the same accuracy as group-specific classifiers.

The work has involved performing a survey of relevant HAR literature regarding both healthy and disabled subjects. With a chapter devoted to the necessary background knowledge in addition to a literature survey, the thesis should serve as an introduction to accelerometer-based HAR for anyone with a background in computer science.

Sammendrag

Slag er en forstyrrelse i hjernens blodtilførsel som kan føre til at hjerneceller dør. Hvert år rammes flere enn 12 000 nordmenn av slag. Mange av disse blir varig bevegelseshemmet som følge av hendelsen, noe som påvirker evnen til å være aktiv og å klare seg på egen hånd. Å vite hvor ofte, hvor lenge og med hvilken intensitet en slagpasient utfører visse vanlige fysiske aktiviteter i sitt dagligliv kan være til nytte i rehabilitering. Fysioterapeuter kan bruke slik kunnskap til å tilpasse behandlingen etter pasientens evner og behov. Forskere kan bruke slike data til å vurdere virkningene av nye behandlingsmetoder.

Vanligvis brukes selvrapportering eller observasjon for å få kunnskap om hvor ofte og hvor lenge en person utfører gitte aktiviteter. Disse metodene fører ofte til unøyaktige resultater på grunn av subjektive feil og observatørens begrensede tid. Systemer for såkalt human activity recognition (HAR), gjenkjenning av aktiviteter hos mennesker, er et alternativ til disse metodene uten slike feil. Ved å bruke maskinlæringsteknikker kan et slikt system gjenkjenne hvilken aktivitet en person utførte på et gitt tidspunkt. Systemet bruker målinger av personens bevegelser, gjort av én eller flere sensorer festet til kroppen, for å utføre oppgaven sin. Tidligere eksempler på målinger, merket med hvilken aktivitet som foregikk under hver måling, brukes for å lære opp systemet. Et system som er lært opp på data fra friske mennesker gir unøyaktige resultater når det skal gjenkjenne aktiviteter hos slagpasienter fordi slagpasienter beveger seg annerledes enn friske mennesker.

Denne oppgavens mål har vært å lage et HAR-system som med høy nøyaktighet gjenkjenner aktiviteter hos slagpasienter. Arbeidet har tatt utgangspunkt i et nytt datasett som er samlet inn fra slagpasienter. Femten slagpasienter deltok i innsamlingen ved å ha på seg fem tre-aksede akselerometre, ett på hvert lår og håndledd og ett nederst på ryggen. Innsamlingsøktene, som tok omtrent en time, fant sted i et laboratorium. Målingene ble siden merket med hvilke aktiviteter som ble utført til hvilke tidspunkter.

Eksperimenter med forskjellige sammensetninger av akselerometre, klassifiseringsteknikker, treningsdatasett og treningsdatamengder har vært med på å forme sluttresultatet: et system som klassifiserer slagpasienters aktiviteter med en ensemble-metode av typen «random forests». I sin ferdige form klarer systemet å gjenkjenne hvilke aktiviteter en slagpasient utfører med en nøyaktighet på 93.4 % ved hjelp av data fra rygg-akselerometret og ett av lår-akselerometrene. Systemet oppnår en nøyaktighet på 94.6 % med ytterligere data fra det andre lår-akselerometret og ett av håndledds-akselerometrene. Et av de senere eksperimentene viser at systemet kan læres opp på et datasett bestående av eksempler fra både friske mennesker og slagpasienter og fortsatt oppnå samme nøyaktighet når det kjenner igjen aktiviteter hos begge disse gruppene som adskilte systemer rettet mot hver enkelt gruppe. Dette åpner for å lage HAR-systemer som ikke sikter seg inn på spesifikke grupper, men som gjenkjenner aktiviteter hos både friske og bevegelseshemmede med samme nøyaktighet som gruppespesifikke systemer.

Arbeidet innebar også en studie av relevant litteratur innenfor HAR-forskningsfeltet med både friske og bevegelseshemmede som tiltenkte brukere. Oppgaven inneholder i tillegg et kapittel som forklarer nødvendig bakgrunnskunnskap. Den bør derfor kunne tjene som en innføring i HAR med akselerometre for enhver leser med bakgrunn i informatikk.

Acknowledgements

First and foremost, I want to thank my supervisors, Kerstin Bach and Helge Langseth, whose feedback and guidance has been invaluable both to this thesis and the specialization project thesis preceding it. The interest they have shown, the time they have spent reading drafts, and the number of feedback meetings they have held has exceeded my expectations. I can wholeheartedly recommend them as supervisors to any student in search for one, just as they were recommended to me by Hans-Olav Hessen and Astrid J. Tessem.

Hessen and Tessem have also been important to my work because of their master's project. My specialization project utilized their activity recognition system to a large degree. Additionally, their thesis has served as an excellent introduction to and reference for the human activity recognition research field. In the same breath, I must mention that the specialization project was a shared effort with Fredrik Gram Larsen, who I think would have been just as enthusiastic as I about continuing our cooperation this semester if the circumstances would have allowed it.

My gratitude goes out to Paul J. Mork and Atle M. Kongsvold at The Faculty of Medicine at NTNU for having gone outside their duties to be of assistance during the literature search, giving both suggestions for and feedback on the medical sections.

I want to thank all the people who gave me permission to re-use their illustrations and photographs, either by publishing their files under permissive licenses or by granting permission when requested. They are credited in accordance with their licensing terms and other wishes, but I want to acknowledge that their graphics do a better job at getting the point across than anything I would have been able to make myself.

The results in this report would not have been the same without the help of Dan Jackson, author of the Timesync synchronization script. After receiving feedback about cases where synchronization was not successful, he quickly made improvements that resulted in near-perfect synchronization.

Lastly, I want to mention Rolf H. Dahl and Alf A. Høiseth at The Department of Computer and Information Science, who granted me access to a very powerful server on which to run my experiments and installed a much needed additional storage unit at my request. The few technical difficulties that appeared were swiftly resolved. I owe them and any other staff member at the department who has had a hand in making these resources available to the students my thanks.

Contents

List of Tables

List of Figures

Glossary

g_0 the standard gravity of the earth, defined to be 9.80665 $^m/_{s^2}$ in the International System of Units (SI). Variations due to local gravity and the effects of the earth's rotation occur, but these are negligible, its value being only 0.5% higher at the poles than at the equator.

application programming interface collection of methods which makes it simpler to develop computer programs for a given piece of software or hardware.

body mass index a measurement of body tissue mass compared to height, equal to weight (in kilograms) divided by height (in meters) squared. Used to give an indication of whether a person has a healthy or unhealthy body weight, with the normal range going from 18.5 $^{kg}/_{m^2}$ to 25 $^{kg}/_{m^2}$.

continuous wave accelerometry the raw accelerometer data format of the Axivity AX3 sensor (AX3).

cycles per second outdated unit of frequency, replaced by hertz (Hz) in the SI.

functional independence measure scale used for assessing functional independence in elderly and disabled individuals.

graphics processing unit computer hardware capable of parallel computations. Can be used to perform faster learning and classification in artificial neural networks.

hertz the unit of frequency in the SI.

International System of Units the most widely used system of measurement, its abbreviation stemming from its French name, *Système international d'unités*.

knee-ankle-foot orthosis assistive leg brace which helps stabilize the leg, ankle, and foot in patients with weakness in the lower extremities.

leave-one-subject-out type of cross-validation in which one subject's data is used as the system's test set and none of this subject's data appears in the training set.

multi-personalization one of the two semipopulation calibration strategies. Calibrates a classifier for the new individual by finding the best sub-model for each activity, which could result in models from different subjects in the sub-model pool being selected.

Opportunity a benchmark human activity recognition (HAR) data set collected for Roggen et al. [2010] which consists of data gathered from multiple types of sensors labelled with four parallel levels of activities, ranging from high-level goals to low-level gestures).

sampling frequency the number of evenly spaced samples delivered by some sensor each second, measured in Hz.

single-personalization one of the two semipoulation calibration strategies. Calibrates a classifier for the new individual by finding the user in the sub-model pool whose entire set of classifiers work best for this individual.

St. Olav's University Hospital a university hospital in Trondheim associated with The Norwegian University of Science and Technology (NTNU).

sub-model in semipopulation approaches, a classifier whose task is to recognize the presence or absence of only one activity.

surface electromyographic sensor attached to the surface of the skin which measures muscle activity.

The Faculty of Medicine and Health Sciences faculty at NTNU.

Timed Up and Go test physical test which involves rising from a chair, performing a three meter walk back and forth before sitting down again.

Trondheim Chronic Stroke stroke HAR data set collected in 2016–2017, first used in this thesis.

Trondheim Free Living non-laboratory data set first used by Larsen and Vågeskar [2016].

Trondheim In-Laboratory laboratory data set first used by Hessen and Tessem [2016].

WAV an intermediary, binary file format generated during the conversion from continuous wave accelerometry (CWA) to comma-separated values (CSV) files.

wearable a sensor which can be worn on the body.

Acronyms

ANN artificial neural network.

API application programming interface.

ARC Activity Recognition Chain.

at affected thigh.

aw affected wrist.

AX3 Axivity AX3 sensor.

BMI body mass index.

BN Bayesian network.

CNN convolutional neural network.

cps cycles per second.

CSV comma-separated values.

CWA continuous wave accelerometry.

DCT discrete cosine transform.

DFT discrete Fourier transform.

DMF The Faculty of Medicine.

ECG electrocardiogram.

FCT fast cosine transform.

FFT fast Fourier transform.

FIM functional independence measure.

GPS Global Positioning System.

GPU graphics processing unit.

HAR human activity recognition.

HMM hidden Markov model.

Hz hertz.

IDI The Department of Computer and Information Science.

KAFO knee-ankle-foot orthosis.

lb lower back.

LEAPS Locomotor Experience Applied Post-Stroke.

LOSO leave-one-subject-out.

lt left thigh.

lw left wrist.

MEMS micro-elecromechanical systems.

MH The Faculty of Medicine and Health Sciences.

ML machine learning.

MP multi-personalization.

NINDS US National Institute of Neurological Disorders and Stroke.

NTNU The Norwegian University of Science and Technology.

RF random forests.

rt right thigh.

rw right wrist.

sEMG surface electromyographic.

SI International System of Units.

SP single-personalization.

SVM support vector machine.

TCS Trondheim Chronic Stroke.

TDIDT top-down induction of decision trees.

TFL Trondheim Free Living.

TIL Trondheim In-Laboratory.

TUG Timed Up and Go test.

ut unaffected thigh.

uw unaffected wrist.

Chapter 1

Introduction

Stroke is a sudden disruption in the blood flow to one or more parts of the brain. This disruption may lead to a death of brain cells. More than 12 000 Norwegians suffer a stroke each year. In the same time period, costs related to the disease amount to 6 billion NOK. The condition may appear in people of all ages, and more than 60 000 Norwegians have suffered a stroke at least once in their life. In the worst case, a stroke incident may lead to death. Those who survive often suffer lasting disabilities, including cognitive deficits and movement problems [NINDS, 1999, Norsk Helseinformatikk, 2017].

Movement disabilities can affect a person's quality of life, as it reduces the disabled person's ability to take part in his or her community. Furthermore, for a stroke survivor, being physically inactive is associated with an increased risk of a repeated stroke. Increasing active time is therefore important both to a stroke survivor's life expectancy and quality of life. Consequently, the main goal for a stroke patient in rehabilitation is to relearn skills that are essential to being physically active, such as independent walking and standing [NINDS, 1999, Nadeau et al., 2013].

Receiving conventional therapy, many stroke patients plateau in their physical abilities within 11 weeks of the incident [Jørgensen et al., 1995]. Thus, new rehabilitation techniques for stroke patients is an active field within medical research. Two methods have conventionally been used to measure the time a person spends performing different activities: self-reporting by the patient and observation by a professional. Both of these methods have known issues, their main problems being misreporting by the patient and the professional's limited time [Roy et al., 2009].

Human activity recognition (HAR) systems present an alternative to self-reporting and observation which does away with these methods' subjective errors and resource problems. Such systems draw upon knowledge from machine learning, a research field within computer science, to make computer programs that automatically recognize what activity a person performed at a given time. Quantitative measurements of the activities' physical effects are used by these systems to perform ther recognition task. The measurements are acquired through one or more sensors. A wide variety of sensors has been used in HAR systems, but the most commonly used sensors are accelerometers. An accelerometer measures the acceleration of a limb or the entire body in one or more directions [Lara and Labrador, 2013, Roy et al., 2009].

Objective physical activity reports are not only useful in research. They can also be of immediate utility to stroke patients and the people around them: Physical therapists could use such reports to adapt exercise programs to a patient's needs and abilities. Tracking progress in active time and physical ability over longer periods could serve to motivate the patient. Caregivers in assisted living could use the reports as an additional source of knowledge about the patient's well-being. The patient's next of kin can use the reports as an assurance that their family member is being sufficiently activized.

It is also possible that a system could issue warnings about possible negative developments in a patient's condition before these would be apparent to a human. This could for instance be done by extracting information about how the patient performs certain activities, using this to compare the patient to previous patients and how their condition developed. Another hypothetical solution would be to find trends in a number of consecutive, week-long recordings, which could be used to project the current patient's development based on previous cases.

This thesis aims to create a HAR system which recognizes the activities performed by a stroke patient based on accelerometer recordings. This requires that medical personnel attach accelerometers to the patient's body which he or she will wear for a period of time, for example a week. During this time, each accelerometer will record how it was affected by the patient's movements. Acceleration in all three spatial directions will be measured at a given rate, possibly as high as a hundred times per second. When the recording is finished, the patient will give these sensors back to the personnel, who will upload the acceleration recordings to a computer. This computer will run the HAR program to get a report about which activities the subject performed and for how long. The types of recognized activities are limited to a set of activities which the system has been taught to recognize by a machine learning algorithm. Examples of these activities are *walking*, *sitting*, and *standing*. The purpose of the activity reports is to aid in research on and rehabilitation of stroke patients.

A new data set has been collected for this thesis, the Trondheim Chronic Stroke data set. The data set consists of hour-long accelerometer recordings of fifteen stroke patients labeled with what activity the subject performed at every point in time. Each patient wore five accelerometers, located on both of their wrists and thighs in addition to the lower back. Data about each subject's height, age, weight, and physical ability has also been recorded. This data set will be used to train a HAR system and test its recognition performance using different numbers of sensors and machine learning approaches. One of these approaches is a recently invented HAR technique, called *semipopulation approaches*. Experiments will investigate how semipopulation approaches affect recognition performance and whether they can be used to detect similarities in activity performance between the patients. The hope is that activity performance similarities will be related to physical ability, so that a similarity between two patients can be used to estimate one patient's physical condition from that of the other.

1.1 Research Questions

- **Goal 1:** Create a HAR system which performs accurate classification for people with motor impairments.
  - **Research question 1:** What sensor placements yield the best results with regards to recognition accuracy?
  - **Research question 2:** What amount of labeled data is necessary in order to make an adequately performing classifier?
- **Goal 2:** Create a HAR system which can be used as a diagnostics tool for medical professionals working with motor impaired patients.
  - **Research question 3:** To what degree is the similarity in movements in individuals indicative of the correlation between the two's health?
  - **Research question 4:** What sensor placements work best in distinguishing individuals based on their physical condition?

1.2 Structure

The remainder of this thesis is structured as follows:

- **Chapter 2: Background Theory.** Presents the background theory necessary to understand accelerometer based HAR for stroke patients, including explanations of the machine learning techniques used and stroke as a disease.
- **Chapter 3: Literature Review.** Explains accelerometer based HAR for healthy subjects and stroke patients, as well as presenting research which is relevant to the interpretation of this thesis' results. The chapter concludes with an explanation of semipopulation approaches, which will be used in the experiments.
- **Chapter 4: The Trondheim Chronic Stroke Data Set.** Describes how the stroke patient data set used in this thesis was collected as well as the properties of the data set.
- **Chapter 5: Methodology.** Elaborates on the design of the HAR system used in this thesis' experiments.
- **Chapter 6: Experiments.** Presents each experiment with an explanation of its motivation, setup, results, and a discussion.
- **Chapter 7: Conclusions.** Summarizes the experiments' findings and concludes with propositions for future work.

Chapter 2

Background Theory

This chapter presents theoretic knowledge and terminology needed to understand the rest of the thesis. Section 2.1 introduces machine learning, section 2.2 explains the decision trees and random forests machine learning techniques, and section 2.3 gives the definitions of some common quality metrics for machine learning systems. Two concepts that are essential to accelerometer HAR, vectors and frequency domain transforms, are explained in sections 2.4 and 2.5. The chapter ends with an explanation of stroke as a condition and stroke rehabilitation in section 2.6.

2.1 Machine Learning

Machine learning (ML) is a sub-field of computer science concerned with making programs that learn from experience, as opposed to being explicitly programmed by humans. The term was coined by Samuel [1959], which examined a computer program's ability to improve its own performance at the game of checkers.

2.1.1 Learning Problems

Essential to any machine learning program is the task at which it is to improve, a so-called learning problem. Mitchell [1997, p. 2] defines a learning problem in the following way:

> A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T as measured by P, improves with experience E.

The author presents a few examples of such learning problems, e.g. recognizing hand-written words in images. In this case, the task T is recognizing hand-written words, the performance measure P is the percentage of words correctly classified, and E is a database with images of hand-written words along with the text which is actually written in the image.

2.1.2 Types of Learning

Russell and Norvig [2010, p. 693–695] divides machine learning (ML) problems into three distinct categories based on what type of feedback is available to the program:

1. **Unsupervised learning**, in which the program gets no explicit feedback and has to learn patterns from the input it is given. The most common unsupervised learning task is clustering, which means separating the data it is given into useful groups.

2. **Reinforcement learning**, in which the program is given positive or negative feedback on its performance after it has completed a complex task. The program itself has to reason about which of its actions caused this feedback to be positive or negative.

3. **Supervised learning**, in which the program has access to a large number of input–output pairs, also called labeled data. The program is to learn a function which maps inputs to outputs.

Some programs cross the boundary between unsupervised and supervised learning by performing so-called **semi-supervised** learning. According to Zhu [2017], there are two categories of semi-supervised learning: inductive and transductive. In inductive semi-supervised learning, which builds upon supervised learning, the program uses unlabeled data in addition to labeled data in order to improve its performance. In the transductive case, which builds upon unsupervised learning, the program is given some restrictions about the unlabeled training data which affects its learning, e.g. by being instructed that two examples must fall into the same or different groups.

This thesis will be concerned with supervised learning.

2.1.3 Supervised Learning

Definition

Russell and Norvig [2010, p. 695] define the supervised learning task as follows:

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

x, the input to the system, is also known as **observations**. The way in which these observations are represented is called the **observation language** [Blockeel, 2010]. This language is some machine-readable format, such as vectors, graphs or sequences. Blockeel [2010] notes that "[p]robably the most used setting in machine learning is the **attribute–value setting**", in which observations are represented as vectors, i.e. ordered sequences of values. For example, if a human is observed, the attributes may be sex, height in meters, weight in kilograms, and age in years, with values given as text and numbers.

y is the true output of the unknown function f and therefore the desirable output of h given x. If y's value is one of a finite set of classes, the learning problem is said to be a **classification** problem. For example, a classification system's task could be recognizing hand-written digits from digital images, and the classes that y could belong to would be the the numbers from 0 to 9. If y's values are numbers from a continuous distribution, the problem is a **regression** problem [Russell and Norvig, 2010, p. 696]. For example, a regression system's task could be to estimate the price at which we should sell a used car based on attributes such as its mileage, which car manufacturer has made it, its number of seats, and so on. In this case, y would be the price at which a car should be sold, given in some currency.

h is a hypothesis, also called a model. It is selected from a **hypothesis space** $\mathcal{H}$. $\mathcal{H}$ is "the set of all hypotheses that might possibly returned by [the machine learning system]" [Blockeel, 2017a], defined by the **hypothesis language**. Examples of such languages are decision trees, rule sets, and neural networks [Blockeel, 2017b]. The space may also be further restricted by a **language bias** (also known as restriction bias), which restricts $\mathcal{H}$ to a certain subset of the hypotheses that can be expressed using the given hypothesis language [Mitchell, 1997, p. 64]. An example of a language bias might be restricting a decision tree hypothesis space to decision trees with maximum depth three.

Learning Algorithms

A **supervised learning algorithm** is any algorithm able to fulfill the supervised learning task by outputting a hypothesis consistent with the observations given to it [Sammut and Webb, 2017a]. The hypotheses output by such algorithms, along with the procedures for getting output from them for a new observation, are called **classification algorithms** or **regression algorithms**, depending on which of the two problems they solve.

Hypothesis spaces are potentially very large. Consequently, many hypotheses within the hypothesis space may be consistent with the training data. To choose between several equally good hypotheses, algorithms need an **inductive bias**. An inductive bias is defined as any assumption about future observations which can not be derived from the training which is used to choose between several consistent hypotheses [Sammut and Webb, 2017d]. Choosing the simplest hypothesis consistent with the training data (often referred to as **Ockham's razor**) is a common inductive bias. This stems from the assumption that a simple hypothesis is more likely to yield good predictions for unseen observations. What constitutes simplicity must be defined within the hypothesis space, e.g. in the decision tree space, trees with fewer nodes can be defined as simpler than trees with more nodes, and in the space of polynomials, lower-order polynomials can be considered simpler than higher-order polynomials [Russell and Norvig, 2010, p. 696].

In supervised learning, it is common to let some of the available observations be unavailable to the learning algorithm. This data is known as a **test set** or an evaluation set. Data available to the learning algorithm is known as a **training set**. An entire set of observations, i.e. both the training and test sets, is known as a **data set** [Sammut and Webb, 2017c].

2.1.4 Correcting Class Imbalance

In many real-world ML problems, the following two things occur: First, the distribution of examples among the classes in the training set is very skewed. Second, the real-world costs associated with misclassifying an observation from a rarely occurring class are much larger than the costs associated with misclassifying a frequently occurring class.

Training sets in which these two events occur simultaneously may have consequences for the output hypothesis. The hypotheses' performance on the training set, measured as the number of correctly classified samples, usually plays a large role in selecting the best hypothesis. If the values of the observations' attributes are distributed in such a way that finding a hypothesis which separates them well is hard, the skewed distribution may lead the algorithm to select a hypothesis which simply outputs a majority class for all samples which fall into the minority class.

To illustrate this, imagine for example using ML to make a smart phone application for elderly people. The application's task is to use the phone's accelerometer to separate two types of events, "regular use" and "fall", and summon medical personnel if it suspects a fall. Misclassifications will have real-world costs: Classifying a "fall" as "regular use" may lead to injury and even death, which has huge associated monetary and emotional costs; classifying "regular use" as "fall" also has a cost, as medical personnel's time will be wasted, but less so that the other way around. Imagine also that the application's training set consists of labeled accelerometer observations collected from many home-dwelling seniors for over a year. Most observations exhibit a low acceleration and are labeled "regular use", but a small number exhibit a very high acceleration. Out of these high-acceleration observations, most have been caused the phone accidentally falling to the ground without the owner falling (labeled "regular use"), but a minority were caused by a dangerous fall (labeled "fall"). During training, the algorithm finds no way to separate the different high acceleration samples and decides upon a hypothesis which classifies all such samples as "regular use", as this leads to the highest accuracy. To say it lightly, using such a hypothesis will probably not lead to a good user experience.

In some cases, the solution to such a problem would be to collect more observations belonging to the minority class. For various reasons, this may not an option in a particular setting, e.g. in the application presented in the last paragraph, it would pose a health risk for the users. This section will explain two ways of dealing with class imbalance in a data set: **cost-sensitive learning** and **sampling**.

Cost-Sensitive Learning

Before explaining cost-sensitive learning, it is useful to introduce the notion of true and false positives and negatives in classification as well as the the concept of cost. Table 2.1 shows how true and false positives and negatives are defined in a two class problem. If the problem had several classes, these statistics would be calculated on a class-by-class basis, regarding samples from the examined class as positive and all other samples as negative.

The basis of cost-sensitive learning is associating a cost with each of these four classification outcomes, C_{TP}, C_{TN}, C_{FP}, and C_{FN}. By definition, positive costs indicate harm, negative costs indicate benefit. Often, the cost associated with the true categories is set to 0, and only misclassifications have an associated cost. The **total cost** of misclassifications for a sample set is equal to the number of samples falling into each of these categories times the cost associated with the category.

Table 2.1: A two-class confusion matrix. Columns show the class output by a classifier, and rows show the actual class of the observation. Green cells indicate correct classification, and red cells indicate misclassification.

		Predicted class	
		Positive	Negative
Actual	Pos.	True positive (TP)	False negative (FN)
	Neg.	False positive (FP)	True negative (TN)

In cost-sensitive learning, the task of the training algorithm becomes to find a hypothesis which minimizes total cost. The algorithm consequently has to be adapted to take the supplied costs into account during training. Exactly how this is done is a matter of implementation. The number of cost-sensitive implementations of an algorithm may be large, e.g. Lomax and Vadera [2013]'s survey of cost-sensitive decision tree induction algorithms found that there were more than 50 different implementations only for decision tree learning. A cost-sensitive implementation of decision tree learning will be explained in section 2.2.2.

Cost-sensitive learning's main disadvantage is that it requires a modified algorithm. Compared to the sampling techniques which will be explained below, its advantage over oversampling is that the computational load on the system is smaller, as the number of samples examined during training remains the same and taking the costs for a sample into account is usually far less expensive than examining duplicates of it. Its advantage over undersampling is that it does not lead to samples in the training set being deleted [Weiss et al., 2007, Ling and Sheng, 2017, Scikit-Learn Developers, 2016].

Sampling

Sampling is changing the distribution of examples in the data set either by duplication or deletion. Two types of sampling are available: oversampling and undersampling. **Oversampling** consists of simply duplicating samples from less frequent classes before training, possibly making them as frequent as the most frequent class. **Undersampling** takes the opposite approach of oversampling and consists of deleting samples from more frequent classes.

Weiss et al. [2007, p. 2], which compared cost-sensitive learning and sampling, explained the effect of oversampling from the findings of Elkan [2001]. In their words, "altering the class distribution [. . .] imposes non-uniform misclassification costs. For example, if one alters the class distribution of the training set so that the ratio of positive to negative examples goes from 1:1 to 2:1, then one has effectively assigned a misclassification cost ratio of 2:1".

An advantage of these techniques is that they can be used with any training algorithm, as they only require that the training set is modified beforehand by simple duplication and deletion operations. Undersampling can also be advantageous if a data set is so large that its size has to be reduced in order for machine learning to even be feasible. A disadvantage of oversampling is that duplication leads to more samples being analyzed during training, and thus the running time increases. Undersampling's main disadvantage is that it leads to samples being lost before training [Weiss et al., 2007].

2.2 Decision Trees and Random Forests

The machine learning method random forests (RF) will be used in this thesis. In order to understand this algorithm, it is useful to first examine decision trees, before explaining RF and its relation to decision trees.

2.2.1 Trees in Graph Theory

In graph theory, a **graph** is a collection of **nodes** connected by **edges**. A graph can be either **directed**, i.e. all of its edges are said to start in one node and end in another, or **undirected**, in which case neither of the two nodes can be said to be the start or end point of the edge.

A **tree** is a graph with two special properties: It is **connected**, meaning that every node in the graph can be reached from every other node through the graph's edges, and **acyclic**, meaning there is one and only one path between any two nodes. In a tree, any node which is only connected to one other node is called a **leaf node**. Every other node is an **internal node**.

Trees are usually **rooted**, which means one of its internal nodes is a designated **root**. The **depth** of a rooted tree is the maximum number of edges from the root to a leaf node in the tree. The significance of a root is most apparent when the tree's edges are directed, in which case the root is the only internal node which has no incoming edges and only outgoing edges[1]. A tree with directed edges and a root is called a **directed rooted tree**. A rooted, directed tree of depth three can be seen figure 2.1.

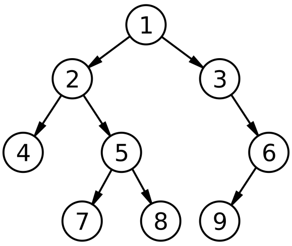


Figure 2.1: Directed rooted tree. Nodes 1, 2, 3, 5, and 6 are internal nodes; the rest are leaf nodes. 1 is the root.

If a tree node A has an outgoing edge to another node B, A is called the **parent** of B, and B is called a **child** of A. The collection of nodes formed by a node's parent, grandparents, and so on up to and including the root is called the **ancestors** of a node. The

[1] This statement is a slight simplification which assumes the tree is a so-called **out-tree**. The root can also have only incoming edges and no outgoing edges, in which case the tree is an **in-tree**. The trees presented in this thesis are all out-trees.

collection of children, grandchildren, and so on are its **descendants** [Cormen et al., 2007, p. 1173–1179].

2.2.2 Decision Trees

The explanation in this subsection is based on Fürnkranz [2017] and Russell and Norvig [2010, p. 697–707].

Structure

Decision trees are directed rooted trees which can be used for classification in attribute–value settings. An example decision tree can be seen in figure 2.2.

Table 2.2: Observations of the weather and a given golf player's decision to play golf in that weather. Taken from Quinlan [1986, table 1].

Outlook	**Temp**	**Humidity**	**Windy**	**Play golf?**
rainy	hot	high	false	no
rainy	hot	high	true	no
overcast	hot	high	false	yes
sunny	mild	high	false	yes
sunny	cool	normal	false	yes
sunny	cool	normal	true	no
overcast	cool	normal	true	yes
rainy	mild	high	false	no
rainy	cool	normal	false	yes
sunny	mild	normal	false	yes
rainy	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
sunny	mild	high	true	no

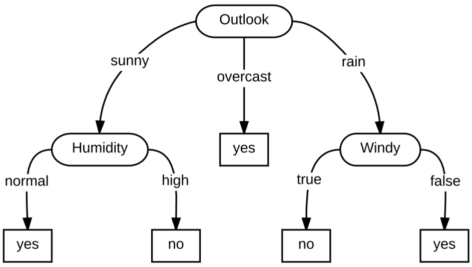


Figure 2.2: A decision tree which can be used to predict whether the golf player from table 2.2 will play golf. Taken from Quinlan [1986, figure 1]

A decision tree's internal nodes correspond to attributes. Each leaf node has a label corresponding to one of the output classes. The edges going out from an internal node correspond the possible values of the attribute specified in the node. All edges seen in figure 2.2 handle discrete valued attributes. It is also possible for edges to handle continuous valued attributes, in which case the edges correspond to some expression, such as "wind_speed > 5".

To classify a new observation using a decision tree, one starts a search at the root. The attribute specified in the root node is examined, and the edge corresponding to the observation's value for this attribute is followed. If the node at the end of the edge is a leaf node, the corresponding classification is output and the search is stopped. Otherwise, the next node is an internal node, the procedure is repeated for the attribute specified by this node.

For example, if we were to classify the first observation in table 2.2 using the decision tree in figure 2.2, we would start by examining the attribute specified by the root node, "Outlook". We would then follow the right edge, because it corresponds to the observation's value for this attribute, "rainy". The next node specifies that the "Windy" attribute should be examined, whose value in the observation is "false". The search is then stopped, as a leaf node has been found, and the leaf node's class "yes" is output as the classification for the observation.

Learning

Decision trees can be built in a top-down fashion, which means it is possible to start construction at the root node, finish the construction of this node, and then move on to constructing its children in the same way. Top-down induction of decision trees (TDIDT) is a recursive function which constructs decision trees in this way, whose pseudocode can be seen in Algorithm 1.

```
function TDIDT($S$)
    $Tree \leftarrow$ new empty node
    if all examples have the same class $c$ then
        Label($Tree$) $\leftarrow c$
    else if no further attributes to split on then
        Label($Tree$) $\leftarrow$ PluralityValue($S$)
    else
        $(A, T) \leftarrow$ FindBestSplit($S$)
        for each test $t \in T$ do
            $S_t \leftarrow$ all examples that satisfy $t$
            $Node_t \leftarrow$ TDIDT($S_t$)
            AddEdge($Tree \xrightarrow{t} Node_t$)
    return $Tree$
```

Algorithm 1: Top-down induction of decision trees (TDIDT), adapted from Fürnkranz [2017, figure 2] and Russell and Norvig [2010, figure 18.5]

Each call to the TDIDT function results in the construction of a new node in the decision tree. The function takes a set of training examples along with their observed class. This set is called S. If this is the first time the function is called (i.e. the algorithm is constructing the root node), S consists of the entire training set. In all other cases, S will be a subset of the training set. The reasons for this will become apparent as the construction of new internal nodes is explained.

As seen in line 3, if all examples in S have the same class, the algorithm constructs a **leaf node** whose label corresponds to this class. Line 5 handles the case where all attributes in the observations have already been examined in some ancestor of the node currently being constructed. In this case, the algorithm constructs a new leaf node with the label output the function PLURALITYVALUE, which returns the most common class in S. If two or more classes are equally common, the function breaks the tie randomly.

If there are still attributes to split on and more than one class in S, the algorithm will be able to construct a new **internal node**. The algorithm uses some function FINDBESTSPLIT to determine the "best" attribute for this node. Exactly what constitutes the best attribute will be explained in the next paragraph. FINDBESTSPLIT returns the best attribute, A, along with a set of tests T. Each such test will correspond to an edge going out from this node. When the attribute values are discrete, the tests are of the form $t \leftarrow (A = v)$, and when values are continuous, they are of the form $t \leftarrow (A < v_t)$. E.g., in constructing the root node in figure 2.2, $A = Outlook$, and $T = (Outlook = sunny, Outlook = overcast, Outlook = rainy)$. The observations in S are then split into several subsets according which test they pass (i.e. their value for this attribute). The algorithm proceeds to construct a node for each such subset S_t.

An Implementation of FINDBESTSPLIT

TDIDT does not specify how to implement FINDBESTSPLIT, but the function should ideally return an attribute which leads to the entire tree generalizing well for unseen examples. As explained in section 2.1.3, simple hypotheses lead to better generalization. To achieve simple trees, FINDBESTSPLIT should select an attribute which minimizes the number of additional splits needed before all leaf nodes for the set S has been constructed.

To know the true answer to which variable will *actually* result in the simplest tree, the function would have to construct all possible trees, down to the last leaf node, and check whether they classify the training set correctly. The number of possible trees grows exponentially with the number of attributes and values, and constructing all possible trees is therefore not feasible for most problems. Practical implementations of FINDBESTSPLIT therefore select a split attribute using some **heuristic**, a rule of thumb which indicates how likely it is that an attribute will result in a simple tree. The remaining paragraphs will explain how FINDBESTSPLIT can be implemented using the information gain heuristic, which relies on measurements of the impurity of the data set.

An **impurity** measure quantifies how skewed the class distribution of the set is: An equal distribution of classes leads to high impurity, while a very skewed distribution leads to low impurity. Two common ways to measure impurity are Gini index, seen in equation 2.1, and information-theoretic entropy, as seen in equation 2.2. In these equations, S refers to the set of training examples which are used to find the split, C refers to the set of classes and S_c refers to the set of examples belonging to class c. Vertical bars, e.g. $|S|$, denote the *cardinality* of the set, i.e. the number of elements in the set. In other words, $|S_c|$ is the number of examples belonging to class c, and $|S|$ is the total number of examples in

S.

$$Gini(S) = 1 - \sum_{c \in C} \left(\frac{|S_c|}{|S|} \right)^2 \tag{2.1}$$

$$Entropy(S) = - \sum_{c \in C} \frac{|S_c|}{|S|} \times \log_2 \left(\frac{|S_c|}{|S|} \right) \tag{2.2}$$

An **information gain** heuristic is a function which takes a data set S along with an attribute A and returns the reduction in impurity by splitting the data set on this attribute. In order to implement FINDBESTSPLIT using an information gain function, we need only to evaluate all attributes which have not been used as a split attribute in any ancestor of the current node using the function. The split attribute returned by FINDBESTSPLIT is the attribute which is found to lead to the highest information gain.

One function which can be used to measure information gain is seen in equation 2.3. This equation expresses the information gain as the impurity of S minus the impurities of each the subsets which will result from performing a split on attribute A. Each subset S_t consists of all classes which have a certain value for this attribute and is weighted by its size relative to the other subsets.

$$Gain(S, A) = Impurity(S) - \sum_{t} \frac{|S_t|}{|S|} \times Impurity(S_t) \tag{2.3}$$

Equation 2.3 tends to favor attributes which can take on a lot of values, which may not lead to good generalization. One way to counter the problem of favoring attributes with many values is to normalize information gain by the attribute's *intrinsic entropy*, i.e. its predisposition for resulting in leaf nodes. A function which does this is seen in equation 2.4.

$$GainRatio(S, A) = \frac{Gain(S, A)}{- \sum_t \frac{|S_t|}{|S|} \times \log_2 \left(\frac{|S_t|}{|S|} \right)} \tag{2.4}$$

Cost-Sensitive Decision Tree Learning

Cost-sensitive learning was presented in section 2.1.4. This subsection will explain how cost-sensitive learning is implemented Scikit-learn's decision tree package[2], which is used for learning the decision trees in this thesis' system. The package's code and documentation were used as the source for the explanation.

Scikit-learn's decision tree algorithm makes cost-sensitive learning possible by introducing one additional piece of information for each sample in the data set: the sample's weight. This is a value which is supplied to the algorithm along with the training set, and each sample's value remains constant for the duration of the algorithm. For the rest of this explanation, let us use w_j to denote the weight of sample j. This weight is a non-negative number which makes the algorithm treat it as if it actually represented more samples. For example, if we have two samples, A and B, where $w_A = 2$ and $w_B = 1$, A actually represents two duplicated samples where B represents one.

[2] https://github.com/scikit-learn/scikit-learn/tree/master/sklearn/tree

How the weights are incorporated into the algorithm can be described as a redefinition of cardinality: Instead of defining cardinality as the *number of samples in a set*, the cost-sensitive algorithm defines cardinality as the *sum of the set's sample weights*. Let us denote the weighted cardinality of a set S as $|S|_w$, as seen in equation 2.5. FINDBEST-SPLIT can then be modified to take these weights into account by substituting all uses of the cardinality operator in equations 2.1 through 2.4 with the use of weighted cardinality operators. It is easy to see how this is equivalent to oversampling, as doubling a sample's weight will have the same effect as duplicating it and keeping the weight constant.

$$|S|_w = \sum_{j \in S} w_j \tag{2.5}$$

Section 2.1.4 explained how sampling could be used to express the relative misclassification costs for different classes. The preceding paragraph explained how sample weights could be considered equivalent to sampling. We will now see how these two ideas can be combined to introduce cost-sensitivity to decision tree learning. Let M_c denote the misclassification cost for class c. If one class has a class weight whose value is twice that of the other class, misclassifying it is considered twice as costly as misclassifying the other class. To make all classes equally costly, M_c should be equal for all $c \in C$.

Let S_c denote set of samples with class c. To achieve the same effect in learning as if each sample in S_c was oversampled, M_c can be distributed equally among all samples in S_c, as seen in equation 2.6. Note that the cardinality in the divisor is not weighted.

$$w_j = \frac{M_c}{|S_c|} \tag{2.6}$$

2.2.3 Ensemble Methods

Random forests, which will be presented in section 2.2.4, is a so-called **ensemble method**. Ensemble methods derive their strength from reducing **variance**. This subsection will present the concepts of statistical bias[3] and variance, their relation to error, and how ensemble methods can reduce the error introduced by variance.

Statistical Bias and Variance

Dietterich and Kong [1995] gave a presentation of the concepts of statistical bias and variance for ML algorithms.

Referring back to section 2.1.3, h is a hypothesis which approximates a true function f. Let us now say that h_S is the hypothesis output by some supervised learning algorithm A for a set of observations S with size m, so that $A(S) = h_S$. The *average hypothesis* of A is the expected result of running A on training sets of size m. Equation 2.7 defines the average hypothesis as the average of l hypotheses trained on independently sampled

[3] Statistical bias is related to the forms of bias previously presented, but is not the same concept.

training sets as l goes to infinity.

$$\bar{h}(x) = \lim_{l \to \infty} \frac{1}{l} \sum_{i=1}^{l} h_{S_i}(x) \tag{2.7}$$

Dietterich and Kong state that A's statistical bias for some observation x is "the persistent or systematic error that the learning algorithm is expected to make when trained on training sets of size m", as seen in equation 2.8. The statistical variance of A for some observation x is "the expected value of the squared difference between any hypothesis $[h_S]$ and the averaged hypothesis $[\bar{h}]$", as seen in equation 2.9.

$$Bias(A, m, x) = \bar{h}(x) - f(x) \tag{2.8}$$

$$Variance(A, m, x) = E\left[(h_S(x) - \bar{h}(x))^2\right] \tag{2.9}$$

Referring to Geman et al. [1992], Dietterich and Kong state that the average error of an algorithm is equal to its bias squared plus its variance, as seen in equation 2.10.

$$Error(A, m, x) = Bias(A, m, x)^2 + Variance(A, m, x) \tag{2.10}$$

These definitions can be used in regression problems. The next section will show these concepts are used in classification.

Classification Bias and Variance

Dietterich and Kong [1995] use the probability of misclassification to define statistical bias and variance for classification problems.

Let us denote hypothesis h_S' probability of misclassifying some observation x as $p_S(x)$. A hypothesis is deterministic, i.e. it will always give the same answer, and therefore this probability is strictly 1 or 0. The definition of misclassification probability is seen in equation 2.11. The authors then define algorithm A's classification *error* for some observation x to be the average probability of misclassification for classifiers trained over all possible training training sets, as seen in equation 2.12.

$$p_S(x) = \begin{cases} 1 & \text{if } h_S(x) \neq f(x) \\ 0 & \text{if } h_S(x) = f(x) \end{cases} \tag{2.11}$$

$$ClassifError(A, m, x) = \lim_{l \to \infty} \frac{1}{l} \sum_{i=1}^{l} p_{S_i}(x) \tag{2.12}$$

Statistical bias and variance in classification are then defined using the error: The bias is 1 if there is more than a 50% chance of misclassifying the sample, as seen in equation 2.13. The variance for x is defined as the difference between the error rate and the bias, as seen in equation 2.14.

$$ClassifBias(A, m, x) = \begin{cases} 0 & \text{if } Error(A, m, x) \leq 0.5 \\ 1 & \text{if } Error(A, m, x) > 0.5 \end{cases} \tag{2.13}$$

$$ClassifVariance(A, m, x) = |ClassifError(A, m, x) - ClassifBias(A, m, x)| \tag{2.14}$$

Using Ensemble Methods

Using the definitions of error, bias, and variance, we can regard one hypothesis (based on an independently sampled subset) as a statistical observation with variance σ^2.

From statistics, it is known that if we have n independent observations, each with variance σ^2, the average of these observations has a variance σ^2/n. This has given rise to the idea of **ensemble methods**: Attaining large collection of hypotheses, each on an independently sampled subset, and averaging their outputs, thereby reducing the influence of variance on the classifiers' output.

Constructing an ensemble requires only that we obtain a set of hypotheses by running some previously known learning algorithm on a number of independently sampled subsets. When performing classification or regression, the output from the ensemble is the average of the outputs from each independent classifier. In the case of **regression**, this will be the average of the hypotheses' outputs. In **classification**, the output is the result of a *majority vote* in which each hypothesis' output is regarded as a vote for its output class. Whichever class gets the most votes is regarded as the ensemble's output.

One practical problem with ensemble methods is the demand for independent training sets. The size of an available data set may be so small that it is hard to divide it into subsets and still expect classifiers trained on these subsets to be accurate. Bootstrap aggregation, **bagging**, is a technique which imitates the effect of independent subsets by drawing random samples from all the available data. To train an ensemble consisting of B classifiers on a training set of size m using bagging, B subsets are created by drawing m samples *with replacement* from the training set for each subset. One hypothesis is trained on each of these subsets, resulting in a collection of B classifiers [James et al., 2013, p. 316].

2.2.4 Random Forests

A single decision tree often exhibits a lot of variance. For example, Dietterich and Kong [1995] examined the average error of 200 decision trees for a simple two-class classification problem and found that nearly half the errors made by the decision trees were attributable to variance.

Random forests (RF), introduced by Breiman [2001], is an ensemble method which aims to reduce the variance of decision trees. It consists of a learning algorithm which results in an ensemble of decision trees. During classification, a classification is obtained from each decision tree as usual, and the output of the ensemble is determined by the majority voting scheme. RF classifiers can also be used to predict the probability of a sample belonging to a certain class, in which the probability of a sample belonging some class is equal to the proportion of the votes that this class got from the decision trees in the ensemble (e.g. 1 out of 10 votes is equal to a 10% probability) [Scikit-Learn Developers, 2016]. An illustration of an RF classifier with three decision trees is seen in figure 2.3.

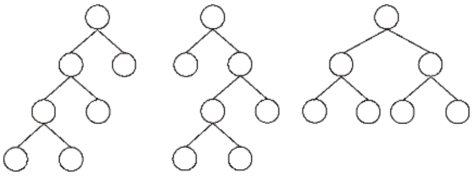


Figure 2.3: Random forests ensemble

The RF learning algorithm requires multiple runs of the TDIDT algorithm with two modifications: The training set supplied to each run of the algorithm is generated using bagging, and the attributes considered when creating a new internal node must be a restricted subset of the attributes. The attributes in this subset must have been chosen at random, hence the name *random* forests.

The reason for restricting the number of considered attributes when creating a new internal node is that some attributes tend to have a very high information gain, even in randomly chosen subsets of the training set. Consequently, if all attributes were considered, the output hypotheses would have a similar structure, i.e. they would *correlate*. Correlation among the classifiers in an ensemble leads to a smaller reduction in variance than would be expected. Restricting the number of available attributes leads to decorrelation of the trees returned by the algorithm and a lower overall variance in the ensemble. How many attributes to consider during a split is a choice of the programmer. The square root of the total number of attributes is a common choice [James et al., 2013, p. 316–321].

An algorithm for constructing an RF classifier can be seen in Algorithm 2. The function RANDOMFORESTS takes a training set S and the number of classifiers which is to be in the ensemble, n. For each classifier, it generates a bootstrap training set Q with the same number of training samples as S using the BOOTSTRAP function. Q is then passed to the function RANDOMTREE. This function is completely identical to the TDIDT function seen in Algorithm 1, except for FINDBESTSPLIT being replaced by RANDOMBESTSPLIT, a version of FINDBESTSPLIT which implements the restrictions described in the preceding paragraph. After all trees have been generated, a classifier ensemble C is returned.

2.3 Quality Metrics

Classification quality is evaluated using measures that show "the degree to which the predictions of a model match the reality being modeled" [Sammut and Webb, 2010a]. These measures build upon the notion of true and false (correct and incorrect) predictions, as shown in table 2.1. In the following formulas, TP and the other abbreviations will serve as shorthands for the total number of samples falling into that class.

Accuracy measures the proportion of correct outputs to the total number of outputs.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.15}$$

Recall, also known as sensitivity, measures the proportion of the actual positives that

```
function RandomForests($S$, $n$)
    $C \leftarrow$ empty array of length $n$
    for $i \leftarrow 1, n$ do
        $Q \leftarrow$ Bootstrap($S$)
        $C_i \leftarrow$ RandomTree($Q$)
    return $C$

function RandomTree($Q$)
    $Tree \leftarrow$ new empty node
    if all examples have the same class $c$ then
        Label($Tree$) $\leftarrow c$
    else if no further attributes to split on then
        Label($Tree$) $\leftarrow$ PluralityValue($Q$)
    else
        $(A, T) \leftarrow$ RandomBestSplit($Q$)
        for each test $t \in T$ do
            $S_t \leftarrow$ all examples that satisfy $t$
            $Node_t \leftarrow$ RandomTree($Q_t$)
            AddEdge($Tree \xrightarrow{t} Node_t$)
    return $Tree$
```

Algorithm 2: Random forests algorithm

were predicted as positive by the model. **Precision** measures the proportion predicted positives that were actual positives [Ting, 2010].

$$Recall = \frac{TP}{TP + FN} \tag{2.16}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.17}$$

The F$_1$-measure is the harmonic mean of precision and recall [Sammut and Webb, 2010b].

$$F_1 = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{2.18}$$

Specificity is “the fraction of negative examples predicted correctly by a model” [Sammut and Webb, 2017e]. High specificity for a class indicates a low probability of a prediction of this class for a sample being wrong.

$$Specificity = \frac{TN}{FP + TN} \tag{2.19}$$

2.4 Vectors

Accelerometers will be explained in section 3.1.4, depending partially on the mathematical and physical concept of vectors, which is related to the vectors described in section 2.1.3, but not the same concept. In physics and mathematics, a vector is a quantity which has a **magnitude** and a **direction**. Force, velocity, and acceleration are examples of vectors.

2.4.1 Vector Decompositon

Any n-dimensional vector can be expressed as a sum of n vectors, each parallel to its own axis in its coordinate system. This is known as a vector decomposition. A 3-dimensional vector decomposition is seen in figure 2.4 [Law and Rennie, 2015].

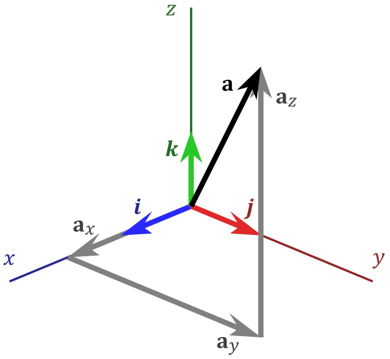


Figure 2.4: Three dimensional acceleration vector a expressed as the sum three vectors, a_x, a_y, and a_z, each of them parallel to an axis in the coordinate system. The vectors i, j, and k are unit vectors, i.e. vectors parallel of length one parallel to an axis.[4]

2.4.2 Relationship Between Axes and Magnitude

The magnitude of a vector v is denoted by $|v|$. Drawing upon the Pythagorean theorem, the square of the magnitude is equal to the sum of the squares of its decomposition. Consequently, the magnitude is equal to the square root of this sum.

The impact of a vector on a given axis depends on magnitude of the force and the angle between the force's vector and the axis. The relationship can be expressed as $|v_i| = |v| \times \cos \alpha_i$, where α_i is the angle between the force and axis i.

2.5 Frequency Domain Transforms

Section 3.1.6 will include a subsection on frequency domain features. This section will explain its theoretical basis, frequency domain transforms.

[3] CC-BY-SA 4.0 User:Acdx, https://commons.wikimedia.org/wiki/File:3D_Vector.svg

2.5.1 Explanation

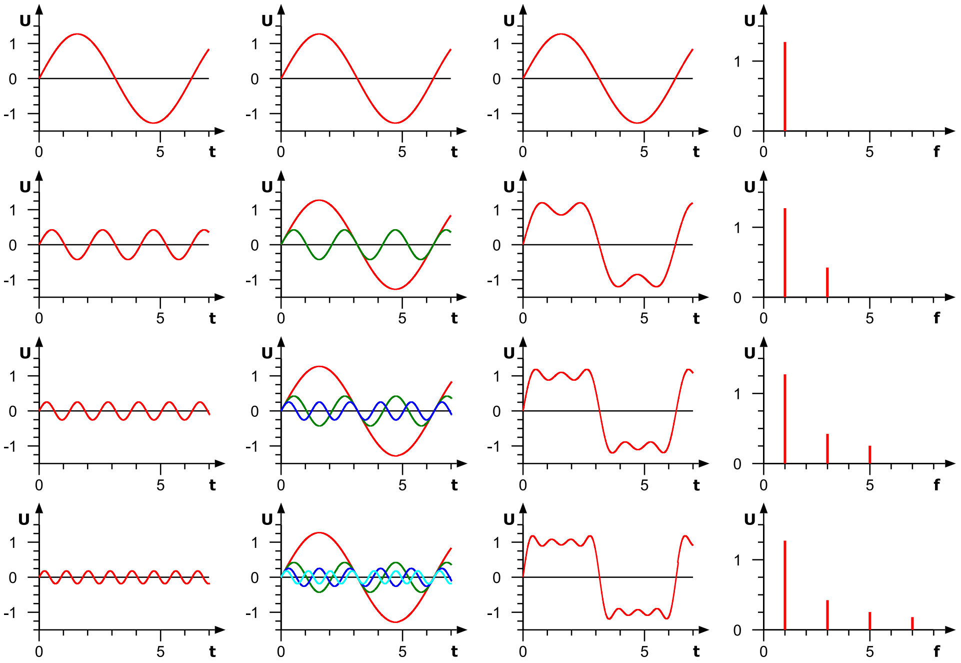


Figure 2.5: Fourier analysis of a square wave with period 2π. One new periodic function is added in each row. The first column shows the individual periodic functions; the second column shows them superimposed; the third column shows the sum of their values; the fourth column shows the value of their coefficients (i.e. their amplitude).[5]

Frequency domain transforms are mathematical operators which can transform a function in the time domain to the frequency domain, i.e. expressing its value as a function of frequency instead of its point in time. These transforms are based on the mathematical concept of Fourier analysis, which states that any real valued function can be expressed as the sum of a possibly infinite set of sinusoidal functions. An illustration of this is seen in figure 2.5.

A discrete Fourier transform (DFT) takes a finite sequence of equally spaced samples and returns the amplitudes of the frequencies present within it. Specifically, applying a DFT to an array containing n real valued samples results in an array of complex numbers $c = [c_0, c_1, \ldots, c_k]$ ($k = \frac{n}{2}$ if n is even and $k = \frac{n+1}{2}$ if n is odd). Each element in c_j is a complex number of the form $x_j + y_j i$ ($i = \sqrt{-1}$). For the purposes of this thesis, it is enough to understand that the absolute value of each of these complex numbers, $a_j = |c_j| = \sqrt{x_j^2 + y_j^2}$, corresponds to the amplitude of a specific frequency in the spectrum. The value in hertz (Hz) of the frequency f_j which a_j corresponds to is a function of the sample spacing d and the number of the samples within the window, n, as seen in

[5] CC-BY-SA 3.0 René Schwarz: https://commons.wikimedia.org/wiki/File:Fourier_synthesis.svg

equation 2.20.

$$f_j = \frac{j}{d \times n} \tag{2.20}$$

The computer algorithms typically employed to transform to the frequency domain are the fast Fourier transform (FFT) and fast cosine transform (FCT). These are implementations of the DFT and the discrete cosine transform (DCT) with computational complexity $\mathcal{O}(n \log n)$, reduced from $\mathcal{O}(n^2)$. The difference between the DFT and the DCT is that the latter only uses cosine functions to describe the time domain function. This thesis will only make use of the FFT.

2.5.2 Use in Signal Processing

In signal processing, frequency domain transforms has been used as the basis for the Nyquist–Shannon sampling theorem, which states that "If a function $f(t)$ contains no frequencies higher than W cps, it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2W}$ seconds apart" [Shannon, 1998, Theorem 1] (cycles per second (cps) is a unit of frequency superseded by Hz). In other words, the sampling frequency determines the highest observable frequency using a given sample rate; the highest observable frequency being half of the sampling frequency.

Looking back at the explanation in section 2.5.1, it is easy to see that this theorem underlies a DFT. For example, a signal sampled at 100 Hz will have sampling spacing $d = \frac{1}{100}$. A 3 second window will contain $n = 300$ samples. Consequently, amplitudes output by a DFT for a signal with this sampling frequency correspond given by the following calculation

$$f = [\frac{0}{\frac{1}{100} \times 300}, \frac{1}{\frac{1}{100} \times 300}, \ldots, \frac{150}{\frac{1}{100} \times 300}] \approx [0, 0.33, \ldots, 50]$$

2.6 Stroke

This section will explain stroke and symptoms of it which are relevant to performing HAR for stroke patients.

2.6.1 The Condition

According to the US National Institute of Neurological Disorders and Stroke (NINDS) [1999], a stroke is a disruption in the blood flow to one or more parts of the brain. This may lead to a death of brain cells if not treated immediately. There are two types of stroke: **ischemic** stroke, which occurs when the supply of blood to the brain is blocked, and **hemorrhagic** stroke, which occurs when a blood vessel bursts and causes bleeding into its surrounding areas. In the US, 9 in 10 stroke incidents are ischemic, the rest being hemorrhagic [Benjamin et al., 2017, p. e375].

Symptoms of a stroke may appear suddenly. Common stroke symptoms, of which one or more may occur, are: numbness of the face, arms or legs; confusion, with difficulty uttering and understanding speech; loss of vision in one or both eyes; loss of balance and coordination, possibly with trouble walking; and a severe headache with no other apparent cause. Medical help should be sought immediately at the suspicion of a stroke, as quick treatment can considerably reduce the potential damage.

Strokes occur in people of all ages and of both genders. Factors such as lifestyle, gender, disease, and age are known to contribute to a higher risk of stroke. According to the NINDS, the most important of these factors are high blood pressure, heart disease, and diabetes.

Damage incurred by a stroke may lead to disabilities affecting both physical and mental health. The NINDS lists five categories of disabilities:

1. **Paralysis:** Complete paralysis (hemiplegia) or weakness (hemiparesis) in one side of the body. Which side is affected depends on what brain hemisphere the stroke occurs in, as the left hemisphere controls the right side of the body and vice versa.

2. **Cognitive deficits:** Thinking, awareness, attention, learning, judgment, and memory can all be negatively affected by a stroke. The patient may be unaware of these deficits.

3. **Language deficits:** Difficulties understanding and producing speech.

4. **Emotional deficits:** Patients may experience difficulties controlling their emotions. Post-stroke depression is common, possibly hindering rehabilitation.

5. **Pain:** Due to damage to sensory regions of the brain, stiff joints or disabled limbs, many stroke patients experience chronic pain.

When referring to stroke patients, it is common to differentiate between acute and chronic stroke patients. Acute stroke patients have suffered a stroke within the last six months, and chronic stroke patients have suffered no strokes within the last six months [Mehrholz et al., 2014, p. 2].

2.6.2 Societal Impact

Worldwide, there were approximately 6.5 million stroke deaths in 2013, accounting for 12% of all deaths. The absolute number of stroke deaths has been increasing since 1990, but adjusting for the increase in overall life expectancy in this period, the age-standardized stroke death rate decreased by 22.5% from 1990 to 2013 [Benjamin et al., 2017, p. e393].

In Norway, which currently has approximately 5 million inhabitants, about 12 000 strokes occur each year (0.2% incidence), two thirds of them in people who have never suffered a stroke before. About 60 000 of the country's inhabitants have suffered at least one stroke (1.2% prevalence), and two thirds of these suffer from disabilities due to it. It is the third most common cause of death, constituting 12% of all deaths in the country. Stroke is also the single disease to require the highest total amount of hospital care days. Costs related to the disease are estimated to amount to 6 billion NOK each year [Norsk Helseinformatikk, 2017].

An extrapolation of self-reported data from the US estimates the prevalence to be about twice as high as in Norway, with 7.2 million adults aged 20 or older having suffered at least one stroke (2.7% prevalence, population of age ≥ 20 being 270 million). The US incidence rate is approximately the same as in Norway, with 795 000 cases each year across all ages

(0.2% incidence, population 320 million). Direct and indirect costs of the disease are estimated to amount to 33.9 billion USD each year [Benjamin et al., 2017, p. e394].

2.6.3 Post-Stroke Rehabilitation

Stroke patients have a 25% chance of a new stroke occurring within the next 5 years of their first stroke. The chances of severe disability and death both increase with each recurrent stroke. Preventing recurrent strokes is therefore essential in preventing death and disability [NINDS, 1999].

There are several ways to prevent and treat a stroke, mainly with medications, surgery, and rehabilitation. Rehabilitation can be either physical or psychological [NINDS, 1999]. The purpose of the HAR system presented in this thesis is to aid in physical rehabilitation. Consequently, only an explanation of physical rehabilitation will be given. How HAR systems can be used in rehabilitation will be presented in section 3.2.

Conventional Therapy

According to the NINDS, physical therapy is "the cornerstone of the rehabilitation process", and its aim is for the patient to "relearn simple motor activities such as walking, sitting, standing, lying down, and the process of switching from one type of movement to another". Conventional therapy usually consists of training, stretching, and exercises administered by a physical therapist [NINDS, 1999]. According to Mehrholz et al. [2014], "Improving walking after stroke is one of the main goals of rehabilitation". A higher level of walking ability is associated with improvements in mobility and participation in the community [Nadeau et al., 2013, p. 378]. Performance on walking tests, such as time required for a 10 meter walk or the meters walked during a 6 minute time interval, is often used as a measurement of this ability [Globas et al., 2012, Mehrholz et al., 2014, Macko et al., 2005].

There is still room for improvement in the rehabilitation of stroke patients. A study of more than 800 acute stroke patients by Jørgensen et al. [1995] showed that as many as 95% of patients who receive conventional rehabilitation treatment plateau in their ambulatory functioning (i.e. walking with or without assisting devices) within 11 weeks after suffering a stroke. A recent study by Paul et al. [2016] used a thigh worn device to assess the physical fitness of community dwelling (i.e. living in private homes) stroke patients compared to healthy subjects of equal gender, age, and body mass index (BMI). The study found that stroke survivors take only half the number steps and spend three more hours being sedentary than their healthy counterparts [Paul et al., 2016, p. 364]. The study concluded that there is a need to find interventions which encourage ambulatory activity in community dwelling stroke survivors.

Treadmill Exercise

An example of an intervention outside of conventional care is treadmill exercise. Based on a review of 44 trials with a total of 2568 patients receiving treadmill exercise as an intervention, Mehrholz et al. [2014] argued that there is increasing evidence that "high-intensity, repetitive, task-specific training might result in better gait rehabilitation". One of the studies examined in this survey, Globas et al. [2012], showed that patients who perform regular treadmill exercise can experience increased benefits compared to stroke patients receiving conventional treatment. Thirty-eight stroke survivors were randomly assigned to conventional care or a treadmill walking exercise regimen. Compared to conventional care, the treadmill walking exercise regimen was more effective in improving cardiovascular fitness, walking endurance, maximum gait speed, balance, self-rated functional mobility, and quality of life. Mehrholz et al.'s review concluded that treadmill exercise is likely lead to improvements in walking speed and endurance, but was not likely to lead to improvements in the ability to walk independently.

Home Exercise

Home exercise programs are another type of intervention. Nadeau et al. [2013] showed that patients who are subjected to a home exercise program (90 minute sessions administered by a professional three times a week for 12–16 weeks) experience the same improvements in walking ability and balance as patients who are subjected to an equal amount treadmill exercise. The result is especially interesting because the home exercises did not involve walking, focusing instead on things like flexibility and upper- and lower-extremity strength. The study's data came from the Locomotor Experience Applied Post-Stroke (LEAPS) trial, a randomized trial of 408 acute stroke patients, making it one of the largest clinical trials of stroke interventions. The LEAPS patients were randomized into three groups, balanced according to walking ability at the start of the trial. The first group received treadmill exercise, the second received home exercise, and the third received no particular exercise, acting as a control group. All three groups were allowed to receive conventional rehabilitation in addition to their in-trial exercise. Walking ability was reassessed at the end of the trial. Both the treadmill and home exercise groups experienced increased walking ability compared to the control group. The authors did not find a correlation between the amount of conventional rehabilitation and walking ability in the treadmill and home exercise groups, but increased amounts of conventional therapy had a positive effect on walking ability in the control group.

Chapter 3

Literature Review

This chapter presents the literature found in the literature search for this thesis. Section 3.1 presents HAR in general, section 3.2 presents HAR for stroke patients and people with stroke-like disabilities, and section 3.3 presents semipopulation approaches, a HAR technique used in this thesis' experiments.

3.1 Human Activity Recognition (HAR)

3.1.1 Definition of HAR

We will now examine HAR as a machine learning problem. Learning problems were presented in section 2.1.1.

Task

Lara and Labrador [2013] provided a comprehensive survey of HAR using wearable sensors, i.e. sensors that are worn on a person's body. The authors define the task of a HAR system 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 $\cup_{j=0}^{r-1} I_j = I$.

The authors also provide a relaxed version of this definition, in which the input data has already been split into time segments of equal duration, reducing the system's task to assigning labels to these segments.

Activities are assumed to be non-simultaneous in both the relaxed and un-relaxed definitions. Lara and Labrador note that this assumption is safe as long as the system's scope is restricted to recognizing **ambulatory activities**, as is often the case. Ambulation, also known as locomotion, is the act of moving through one's physical environment by performing activities such as walking, jumping, standing, and sitting. By their nature, these activities seldom occur simultaneously. If a system is to identify activities that *may* occur simultaneously, e.g. talking on the phone and walking, this assumption may no longer hold. A number of HAR research papers, such as Gu et al. [2009], Ye et al. [2015], and Helaoui et al. [2011], have made efforts to recognize concurrent and interleaved activities.

Note that in the HAR literature, the term **model** is often used to describe the same concept as is described by the term hypothesis in ML. A model is the result of training a system on a given set of training data.

Experience

With regards to *learning type* (see section 2.1.2), the majority of HAR systems examined in Lara and Labrador [2013] are supervised learning systems. This means the systems require training data labeled by humans to perform their task. Liu et al. [2016]'s survey of HAR using smart phones, a sub-domain of wearables, found that supervised learning methods were still dominant three years later.

HAR systems that perform semi-supervised learning exist: Maekawa and Kishino [2013] used semi-supervised learning to build personalized models using labeled time segments gathered from physically similar users. Guo et al. [2016] presented a system able to recognize activities falling outside of the activities in its training set by clustering time segments displaying similar characteristics. Huynh et al. [2008] used an unsupervised method which examined high level annotations, e.g. "being at the office", along with acceleration sensor data to find new, lower-level patterns that could afterwards be identified by a human as "going to the toilet" and "talking at a whiteboard". Lara and Labrador's discussion of semi-supervised HAR concluded with stating that the research area had "not reached maturity", as all semi-supervised systems that could demonstrate any performance benefits when compared to supervised systems either had high demands for computational power, memory or input data.

Another important influence on a system's experience is its *intended users*, which will affect how many and which people will provide data for the system to learn from. In the words of Lara and Labrador, "[s]ome authors claim that, as people perform activities in a different manner (due to age, gender, weight, and so on), a specific recognition model should be built for each individual". This requires that every person who is to use the system also provides training data, which will be used to train this user's model. Other research efforts focus on building models for larger groups of people, which will not require each new user to provide training data, but which requires that the system is trained on a large set of training data, gathered from many different people. Examples of research that investigate the differences between these approaches will be discussed later in this thesis, two of them being Bao and Intille [2004] (presented in section 3.1.8) and Lonini et al. [2016] (presented in section 3.2.2).

Performance Measure

As in all ML, HAR systems learn from a training set and are evaluated on a test set, the training and test sets being disjoint. The contents of the test set should be so that the system's performance on the test set, after learning from the training set, is representative of how the system will perform in new situations [Bulling et al., 2014, section 3.6.5]. Thus, the approach to evaluation will be affected by the system's intended users: If the model output by the system is intended to be used only for recognizing activities from one user (which we from now on will refer to as a *personalized* model) the training and test sets should consequently be collected from the same user. The test data may have been collected at some other time than the training data or may simply be a subset of all the user's available data. Preferably, this process should be carried out for several users [Lara and Labrador, 2013, p. 1196]. If the system is expected to recognize activities without training data from new users (which we will from now on refer to as a *non-personalized* model), the system is usually trained and tested through some method of **cross-validation** on the available data. There are two kinds of cross-validation: ***k*-fold** cross-validation and **leave-one-subject-out (LOSO)** cross-validation. In *k*-fold cross validation, the available data set is split randomly into *k* equally sized subsets, each such subset called a *fold*. During training, all folds except one is combined into a training set, and the remaining fold is used for evaluation. Each fold is left out from training in turn, and the system's performance is summarized across all folds [Sammut and Webb, 2017b]. LOSO regards each user's available data as a fold. Each user is left out from the training set and used as a test set in turn [Lara and Labrador, 2013, p. 1196][Bulling et al., 2014, section 3.6.5]. If a system is not expected to have access to training data from new users, it could be argued that LOSO is more representative of the system's actual performance than *k*-fold cross validation, in which data from users in the test set is also present in the training set [Mannini et al., 2016, p. 2].

When summarizing the system's performance on the test set, the accuracy, recall, precision, and F-score metrics explained in section 2.3 are the most used, according to Lara and Labrador [2013]. Other metrics, such as Kappa statistic and ROC curves, are also used.

Summary of HAR as a Learning Problem

Summarizing and tying into the definition of learning problems from section 2.1.1, a HAR system's task T is to assign time segmented data with labels that correspond to the activity being performed at that time. Its experience E is previously seen time segmented data, usually labeled by humans. Performance P can be measured in a number of ways, of which accuracy, F-score, precision, and recall are examples.

3.1.2 Structure of HAR Systems: The Activity Recognition Chain

Supervised HAR systems commonly perform their task by executing certain sub-tasks in a given sequence. Bulling et al. [2014, figures 1 and 3] made an effort to capture the data flow in such systems in their tutorial on HAR using inertial sensors, resulting in the

Activity Recognition Chain (ARC), as seen in figure 3.1. A similar data flow diagram is seen in Lara and Labrador [2013, figure 1].

The ARC consists of the following steps:

1. **Data Acquisition:** Data is collected from subjects through sensors. Sensors will be explained in the next section. The raw data may need to be converted to some other format before it can be used by a system, and data from several sensors may also need to be synchronized before it is used. Conversion and synchronization in this thesis' system will be explained in section 5.1.

2. **Segmentation:** Data processed in the previous step is segmented into windows, which contain sensor data for a given time segment. These windows may be of equal length or have different lengths. In the latter case, the system tries to segment the data at points where it is likely that a change in activity occurs.

   Windows may be disjoint or overlap. If windows overlap, the start of the next window will be somewhere inside the current window. When extracting windows with equal lengths, overlap is specified as a percentage of the window length. The start of the next window will occur at $(1 - overlap) * duration$. For example, the start of the next window when extracting 1 second windows with 80% overlap will be 0.2 seconds after the start of the current window.

3. **Feature Calculation:** Features are calculated for the windows. A feature is some function of the data in the windows, some of the simplest being mean and standard deviation. This is explained in section 3.1.6.

4. **Modeling:** Occurs only when learning a new model. The system's supervised learning algorithm is used to learn a new model from the window features along with their corresponding labels, generated by humans. Some windows, representing some portion of the entire data or all data from a given subject, are set aside for evaluating the system's performance in the next step and are not used for learning.

5. **Classification:** The system's model is used to obtain activity classifications from the window features. When learning a new model, the system's output for windows which have not been used in learning is compared to their true values (human generated labels). A report about the system's performance, containing statistics such as those seen in section 2.3, is generated.

3.1.3 Data Acquisition Sensors

This section will present sensors and two sensor taxonomies.

Definition of a Sensor

HAR systems use quantitative measurements of attributes of a human's activities to perform classification. Any instrument which can acquire such measurements can be called a **sensor**, common examples of which are accelerometers, gyroscopes, microphones, and cameras.

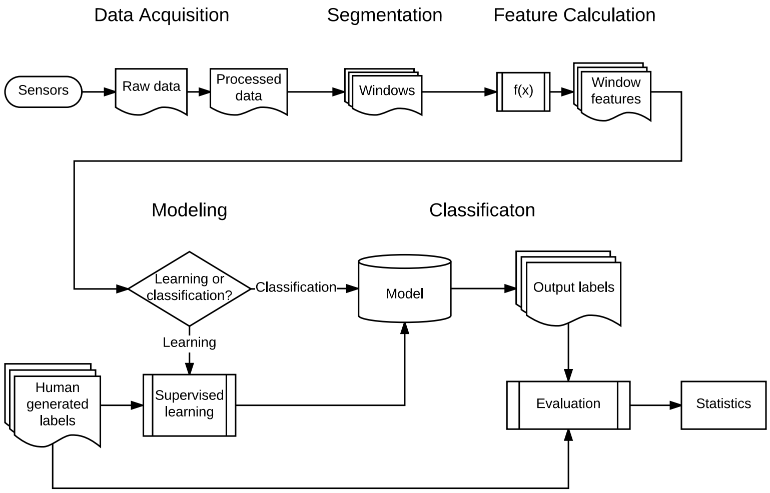


Figure 3.1: The Activity Recognition Chain, modified from Bulling et al. [2014, figures 1 and 3]

Commonly, a sensor a registers one measurement of each of its attributes at intervals evenly spaced in time. The number of measurements for an attribute each second is referred to as the *sampling frequency*. The unit of sampling frequency is Hz. Sensors differ in their sampling frequency: Some sensors, such as Global Positioning System (GPS) trackers or light sensors, may have a sampling frequency less than 10 Hz, while others, such as accelerometers, may have sampling frequencies as high as 100 Hz. [Bulling et al., 2014, p. 9].

Two sensor taxonomies which complement each other will now be presented. Figure 3.2 shows these taxonomies.

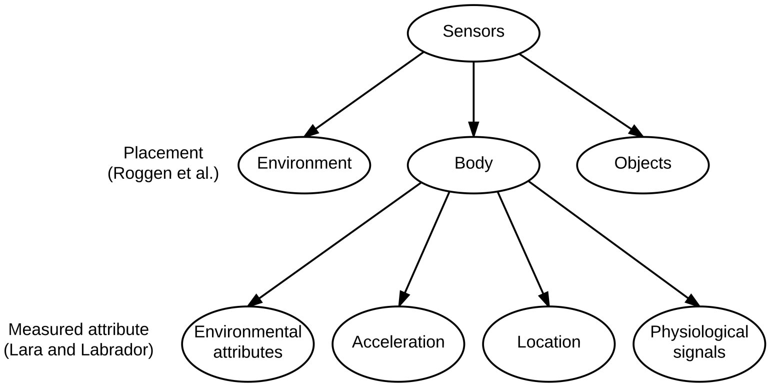


Figure 3.2: Roggen et al. [2010] and Lara and Labrador [2013]'s sensor taxonomies.

Roggen et al.'s Placement Based Taxonomy

A sensor taxonomy was devised by Roggen et al. [2010] in collecting the Opportunity data set, which contains data from 72 sensors of 10 different types. The study's purpose was to collect data from subjects performing household activities in a faux apartment, labelling them with annotations ranging from high level goals, such as "making breakfast", down to ambulatory activities and small manipulative gestures, such as "close drawer".

The taxonomy organizes sensor types into three categories by their placement, either in the subject's **environment**, on its **body** or on **objects** they interact with. Sensor types may fall into one or more of these. For instance, accelerometers can fall into all of these categories, as they can be placed on a subject's body to register attributes of ambulation and manipulative gestures, but also on objects and in the environment to measure interactions with these. Accelerometers will be thoroughly explained in section 3.1.4. Some sensors fall distinctly into one category, an example of this being infrared proximity sensors, which can be used to gain knowledge about a subject's location when placed in the environment.

Bulling et al. [2014, table 3] extended this taxonomy with further examples, the most notable addition being GPS sensors to the environment and body categories. Its omission from Roggen et al.'s paper may be seen as a consequence its scope being indoor living, where GPS sensors are not accurate enough to reliably measure location changes.

Lara and Labrador's Attribute Based Taxonomy

Lara and Labrador [2013] proposed a taxonomy for wearable sensors. As all such sensors are placed on the body, the taxonomy can be seen as a sub-taxonomy of Roggen et al.'s body category.

This taxonomy organizes the sensor types into four categories by what attributes can be derived from them. The categories are: **environmental attributes** such as light intensity and audio level, which can be measured with light sensors and microphones; **acceleration**, related to manipulative gestures and ambulation as measured by gyroscopes and accelerometers; **location**, relating to geographical location as measured using GPS sensors or Wi-Fi networks [Liao, 2006, p. 13-15]; and **physiological signals**, which measure vital data such as heart rate and skin conductivity.

3.1.4 Accelerometers

This section will explain how accelerometer sensors work, the utility of wearable accelerometers, and how the amount of sensors and their placement has previously been shown to affect classification quality.

Construction

Lara and Labrador [2013, p. 1194] claim three-axis accelerometers are "perhaps the most broadly used sensors to recognize ambulation activities", giving their low cost and power requirements as reasons for their widespread use in research. Cheung et al. [2011, p. 999]'s survey of accelerometer HAR found that the number of axes in the accelerometers used in research has increased gradually: Most research before 2000 used one-axis accelerometers, after which two-axis accelerometers became the norm. Since 2008, nearly all research has used three-axis accelerometers.

The accelerometers used in this thesis' data set are Axivity AX3 sensors (AX3s), three-axis micro-elecromechanical systems (MEMS) accelerometers. MEMS accelerometers come in two variants: piezoresistive and capacitive [Albarbar et al., 2009, p. 791]. The AX3 is a capacitive accelerometer[1]. One-axis capacitive accelerometers are based on having a proof mass suspended above a conductive electrode. The electrode is connected to circuitry which measures the current flowing through it. When an external force affects the device, the mass is displaced in relation to the conductive electrode, which affects the flow of current through the electrode. The acceleration can be derived from the measured change [Yazdi et al., 1998, p. 1642].

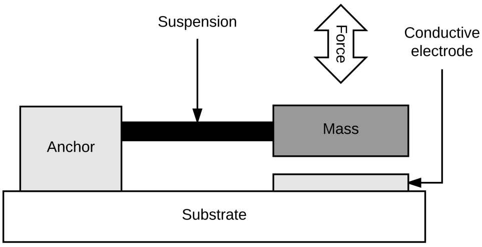


Figure 3.3: Cross section of a vertical capacitive accelerometer. Based on Yazdi et al. [1998, figure 2a].

Section 2.4 explained vectors and how acceleration is an example of a vector quantity. Drawing upon the concept of vector decomposition, it is possible to construct accelerometers for more than one axis by composing single-axis accelerometers, each measuring the acceleration along an axis perpendicular to the others.

Accelerometers differ mainly in two ways: By sampling frequency (see section 3.1.3) and sensitivity, i.e. the range of acceleration it is capable of recognizing. Sensitivity is commonly specified in $\pm g_0$, a unit of acceleration equivalent to the acceleration due to Earth's gravity.

Referring to the work of Maurer et al. [2006], Lara and Labrador state that accelerometers with sampling frequency 20 Hz and sensitivity $\pm 2\ g_0$ are sufficient to recognize ambulatory activities. The requirement for sampling frequency is not surprising, as 20 Hz is sufficient for detecting signals up to 10 Hz according to the Nyquist-Shannon sampling theorem (see section 2.5.2), and it has been shown that at least 98% of the energy caused by human movements is attributable to frequencies of 10 Hz or less [Antonsson and Mann, 1985, p. 44]. The requirement for acceleration is more surprising, as acceleration with a magnitude as high as 12 g_0 has been observed at the ankles during running [Bhattacharya et al., 1980, p. 883].

[1] The sensor being capacitive is not specified in the official product documentation at http://axivity.com/files/resources/AX3_Data_Sheet.pdf. The information comes from correspondence with Axivity.

Types of Accelerating Forces

The acceleration experienced by wearable accelerometers during ambulatory activities have two main causes: **gravity** and **body movements**. Measurements of acceleration during **transportation** has also been used, both alone and along with measurements from other sensors, to distinguish different modes of transportation (such as riding buses, bikes or elevators) [Reddy et al., 2010, Khan et al., 2014], but these activities are outside the scope of this thesis.

As explained in section 2.4, an accelerating force's impact on an axis depends on the angle between the axis and the force. From the point of view of the Earth's surface, **gravity** has a constant magnitude (g_0) and direction (downwards). Its impact on a sensor's axes is therefore dependent only on the sensor's orientation, i.e. the orientation of the body part it is attached to. Veltink et al. [1996] showed that, using two one-axis accelerometers attached to the torso and one leg, it is possible to distinguish standing, sitting, and lying using only from knowledge about a sensor's orientation, derived from its gravity component.

One method to extract the gravity from an accelerometer signal is using a **low-pass filter**. Khan et al. [2014, p. 4] used a built in low-pass filter method in the Android operating system's application programming interface (API) to extract the gravity component from a smart phone accelerometer, which is shown in equation 3.1.

$$g_t = \alpha * g_{t-1} \times \alpha + (1 - \alpha) \times a_t \tag{3.1}$$

The method calculates gravity's influence on a given axis at time t as a function of the last calculated gravity g_{t-1} and the currently registered acceleration a_t [Google, 2014]. Khan et al. found that the API's built in value for α, 0.8, worked best for their purpose, which was recognizing ambulatory activities as well as some household and transport activities (e.g. vacuuming and driving a car).

The acceleration caused by **body movements** is dependent on placement as well as orientation, as only some parts of the body may be involved in a specific movement. E.g. a single sensor placed on the wrist may serve well to distinguish manipulative gestures, such as typing on a keyboard or making breakfast, but may not be appropriate when recognizing ambulation, as the readings from an activity like accidentally swinging one's arm may look similar to the readings from ordinary walking [Lara and Labrador, 2013, p. 1194].

A Practical Example

Figure 3.4 will serve to explain how orientation and placement affects how accelerating forces are registered different on different sensors and axes. In the illustration, the Y-axis of the sensor attached to the subject's side is most affected by gravity, as this axis is perpendicular to the ground. In the thigh sensor, the force is distributed between the Y- and Z-axes at the current time. This distribution will shift as the subject moves his thigh and the axes' orientations change relative to the direction of the force of gravity. Should the subject change the orientation of his torso, e.g. by lying down or bending, the impact of the force will shift to the other axes.

As for body movements, should the subject start accelerating linearly forwards (e.g. by walking or running), the X-axis of the torso sensor and the Z-axis of the thigh sensor

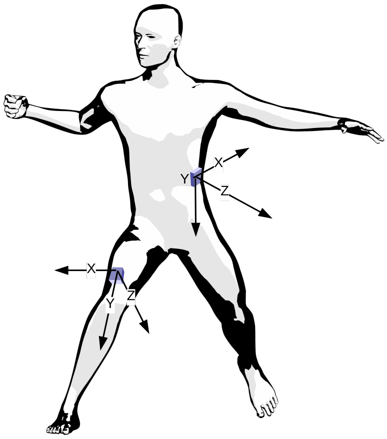


Figure 3.4: Example of sensors attached to body with axis orientations superimposed. Placements and axis orientations have been chosen for the sake of the example and differ from the placements in the data set.

will be most affected by this. The Y-axes of both sensors will experience periodic impacts from the ground's normal force. Additionally, the thigh sensor's Y-axis will experience a centripetal force from the thigh rotating around the hip joint.

3.1.5 Impact of Multiple Accelerometers and Different Placements

Some papers have made an effort to investigate how the number of accelerometers and their placement impacts classifying certain activities. As mentioned in the preceding section, an accelerometer's utility in classifying a certain activity depends on whether it is placed on a body part affected by the activity. Consequently, the set of activities which are to be examined affects the overall evaluation of the sensors. E.g. Bao and Intille [2004] (which will be presented in section 3.1.8) found that sensors placed on the dominant wrist and thigh were sufficient to recognize household activities in addition to ambulatory activities. As this thesis is concerned with ambulatory activities, this section will only present papers which were concerned exclusively with ambulatory activities.

Cleland et al. [2013] performed a rigorous examination of the optimal accelerometer placements for recognizing seven ambulatory activities: *walking*, *jogging* (on a treadmill), *sitting*, *lying*, *standing*, *ascending stairs*, and *descending stairs*. Sensors were placed on six different body parts (chest, lower back, foot, hip, thigh, and wrist). Training and testing using every possible combination of sensors from one to six sensors was performed. It should be noted that the study had only eight participants, all healthy males aged between 24 and 33 years, and that the training and testing procedure was a 10 fold cross-validation using four different classification algorithms. Apparently, no measures were taken to prevent same-subject-data from being present in both training and testing sets, which may explain the high accuracies achieved: In evaluating which single sensor is best for classification, the authors conclude that the best location for a single sensor was the hip, achieving 97.8% accuracy. The thigh was only slightly worse, achieving 96.8% accuracy. The worst locations were the foot and the wrist (still achieving above 95% accuracy) [Cleland et al., 2013, p. 9193]. The report also evaluates all possible combinations of these sensors, using 2 to 6 sensors. Their findings indicate that there is no significant increase in accuracy after increasing the number of sensors from one to two (although a slight increase was shown from 2 to 3 sensors). Also, the differences between any combination of two sensors were insignificant, with the system achieving about 97.5% accuracy for any such combination.

Pannurat et al. [2017] performed a similar evaluation as Cleland et al. using a gender based data set collected from 12 healthy subjects from 23 to 45 years. The sensor positions examined were: upper arm, wrist, ankle, chest, waist side, waist front, and thigh. The labeled activities were: *sitting*, *lying (prone)*, *lying (supine)* (i.e. on the back), *lying (left side)*, *lying (right side)*, *standing*, and *walking*. Unfortunately, this report only examined the use of one sensor at a time, but an advantage of its evaluation is it used a six-fold cross validation approach, leaving two subjects out in every round. Consequently, its results can be assumed to give a more realistic impression of the classification quality for subjects outside the data set. Evaluation was carried out using seven different classification algorithms. In presenting the best accuracies achieved by any classifier for each sensor position, the thigh was shown to be the most accurate (99.0%), closely followed by the chest and the waist side (both above 98%). Once again, the wrist was shown to be the worst position (80.6%), barely beaten by the upper arm (80.8%). Contradicting Cleland et al.'s results for the foot, the ankle performed fairly well (90.7%) [Pannurat et al., 2017, table 3].

From the results of Cleland et al. and Pannurat et al., it seems clear that the accelerometers placed on the wrists are not well suited for classification of ambulatory activities, while accelerometers placed on the hip, torso, and thigh all yield good results. For optimal accuracy, two, possibly three, sensors should be used.

3.1.6 Feature Extraction

This section will explain the purpose and definition of feature extraction followed by explanations of structural, time domain, and frequency domain features.

Purpose

As stated in section 3.1.1, the task of a HAR system is to use classification to assign activities to time segments. As seen in figure 3.1, these time segments are the output of segmenting the processed sensor data. A time segment contains all the quantitative measurements from all sensors within its time period.

Lara and Labrador [2013, p. 1196] argues that if the processed sensor readings from each time segment were to be used as input to a learning algorithm (see section 2.1), the system's task would be "nearly impossible", as the quantitative measurements of two similar activities could be very different. To make the system's learning task simpler, feature extraction must be applied. The purpose of this process is "filtering relevant information and obtaining quantitative measures that allow signals to be compared". Bulling et al. [2014, section 3.3] argues that such features should be "discriminative for the activities at hand", i.e. be sufficient to separate the different activities.

Definition

Bulling et al. [2014, equation 4]'s definiton of feature extraction is shown in equation 3.2, where $\mathcal{F}$ is a set of functions, $\mathbf{D}'$ is the pre-processed sensor data, $\mathbf{w}_i$ defines the start and end of the i-th time segment, and $\mathbf{X}_i$ is a vector containing the features extracted for the time segment.

$$\mathbf{X}_i = \mathcal{F}(\mathbf{D}', \mathbf{w}_i) \tag{3.2}$$

The functions in $\mathcal{F}$ can be either **statistical** or **structural**. The former relies on extracting information from the data based on statistical features, while the latter is based on extracting features based on relationships among the quantitative measurements [Olszewski et al., 2001, p. 2–3][Bulling et al., 2014, p. 9]. Statistical features are usually either time domain or frequency domain features [Lara and Labrador, 2013, p. 1196].

Time Domain Features

Time domain features are statistical features that can be extracted from the quantitative values of the sensor data without first transforming it to some other domain. Many time domain features can be extracted from a segment using one or more un-nested loops, giving the operations a computational complexity of $\mathcal{O}(n)$, n being the number of samples in each time window. Others, such as finding the median or interquartile range, may involve sorting the values, making the operations $\mathcal{O}(n \log n)$ [Cormen et al., 2007, Chapters 5 and 9]. Table 3.1 shows the time domain features used in this thesis.

Extracting time domain features from accelerometer data has also been shown to use less than half the energy needed for frequency domain features while yielding accuracies above 90% for ambulatory activities, making them well-suited for online classification in battery-powered devices such as smart phones [Khan et al., 2013, figure 8][Khan et al., 2014, table 4].

An extensive listing and evaluation of time domain features which can be extracted from an accelerometer, and a gyroscope can be found in Capela et al. [2015]. The authors used filter methods to evaluate 76 time domain features extracted from a smart phone accelerometer and gyroscope by their significance in classification for three different populations (healthy adults, stroke patients, and elderly subjects). As some of these rely on information from gyroscopes, not all are applicable to accelerometer-only HAR.

Frequency Domain Features

Frequency domain features are statistical features that require the sensor data within a window to be transformed to the frequency domain before extraction. Frequency domain transforms were explained in section 2.5, and as said there, such a transform has complexity $\mathcal{O}(n \log n)$. Extracting frequency domain features is therefore more computationally expensive than extracting most time domain features. Table 3.2 shows the frequency domain features used in this thesis.

The motivation for using frequency domain features is that the characteristics of an activity as registered by a sensor may be periodic [Bulling et al., 2014, table 1]. For example, walking and running at a steady pace are both periodic when registered by an

Table 3.1: Time domain features

Name	Definition	Description
Mean	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$	Arithmetic mean of values for an axis.
Standard deviation	$s_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$	Root of the uncorrected variance (the average squared distance from mean).
Skewness	$b_x = \frac{\frac{1}{n}\sum_{i=1}^{n}(x - \bar{x})^3}{s_x^3}$	How "skewed" the distribution of values are around the mean.
Magnitude maximum, mean, and standard deviation	$m_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ $\bar{m}, s_m, \max(m)$	The maximum, mean, and standard deviation of the magnitude of the signal.
Zero crossing 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 crossing rate	$mcr_x = \frac{\sum_{i=2}^{n} \lvert sgn(x_i - \bar{x}) - sgn(x_{i-1} - \bar{x}) \rvert}{2(n-1)}$	Like zcr, but number of times the mean is crossed.
Root mean square	$rms_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$	The root of the mean of the squared values.
Energy	$E_x = \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \quad E_{total} = \frac{1}{3N}(E_x + E_y + E_z)$	A measure of the signal's strength.
Median	$\tilde{x}_{odd} = x_{\frac{n+1}{2}}, \quad \tilde{x}_{even} = {}^1\!/{}_2(x_{\frac{n}{2}} + x_{\frac{n}{2}+1})$	Value(s) separating the sorted values into two equal halves.
Range	$\max(x) - \min(x)$	Difference between maximum and minimum of a sequence
Interquartile range	$iqr_x = Q_{3x} - Q_{1x}$	A quarter of the values in the sorted sequence x are below or equal to Q_{1x}, and three quarters below or equal to Q_{3x}
Correlation	$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$	Pearson's product-moment coefficient. The degree of linear dependence between two series.
Hadamard product mean, standard deviation, and maximum	$xy = x \circ y, \quad xyz = x \circ y \circ z$ $\overline{xy}, s_{xy}, \max(xy); \quad \overline{xyz}, s_{xyz}, \max(xyz)$	The Hadamard product is the element-wise multiplication of the entries in a vector, resulting in a vector of equal length.

accelerometer, as the acceleration perpendicular to the ground will periodically increase and decrease. This is due to the normal force exerted by the ground as the wearer extends and contracts his or her legs.

It is possible for two periodic signals with different periods to exhibit similar time domain features, e.g. two sine waves of equal amplitude will have the same mean and standard deviation although their period is different. Knowledge about the period of a signal can thus help to distinguish activities which would look similar if examined only in the time domain.

Table 3.2: Frequency domain features

Name	Definition	Description
Mean amplitude	$\bar{a} = \frac{1}{k} \sum_{j=0}^{k} a_j$	The arithmetic mean of the amplitudes.
Amplitude standard deviation	$s_a = \sqrt{\frac{1}{k} \sum_{j=0}^{k} (a_j - \bar{a})^2}$	The root of the uncorrected variance for all the amplitudes.
Maximum amplitude	$\max(a)$	The maximum amplitude.
Median amplitude	$\tilde{a}_{odd} = a_{\frac{k+1}{2}}$ $\tilde{a}_{even} = 1/2(a_{\frac{k}{2}} + a_{\frac{k}{2}+1})$	The value which separates the sorted amplitudes into two equally sized halves. If the number of amplitudes is even, it is the arithmetic mean of the two values which separate the values.
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	$f_{\mathrm{argmax}_j\, a}$	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.

Structural Features

Many surveys found in the literature search for this thesis mention structural features, among them Bulling et al. [2014], Liu et al. [2016], and Lara and Labrador [2013]. All of these refer to Olszewski et al. [2001] for the definition of structural features, which explains structural pattern recognition for electrical signals. Only the latter was found to refer to a work in which structural features were actually extracted and used in HAR. This section will explain the utility of structural feature extraction by presenting and discussing the findings about structural features presented in this work, Lara et al. [2012].

Lara et al. [2012] evaluated how structural features could complement time and frequency domain features when distinguishing five ambulatory activities (*walking*, *running*, *ascending*, *descending*, and *sitting*) collected from eight subjects. The structural features were extracted from vital data, namely heart rate, respiration rate, breath amplitude, skin temperature, posture (inclination), and electrocardiogram (ECG) amplitude, all collected by one sensor attached to the chest. Time and frequency domain features were extracted from acceleration data collected using the same device. Structural features were extracted by fitting linear, polynomial, exponential, and sinusoidal functions to each such measurement within each time frame. The authors' argument for extracting structural features instead of statistical features was that "vital signs have much lower variability than acceleration signals". For example, when a subject starts running, the user's pulse will not increase before several seconds have passed. In the same manner, when a subject stops running, his heart rate does not decline before several seconds have passed. Statistical features extracted from the same measurements would not serve to be discriminative [Lara et al., 2012, section 3.2.2].

To argue that *statistical* features extracted from vital data are not discriminative, Lara et al. refer to a study by Tapia et al. [2007], which sought to recognize ambulatory and physical activities performed at different intensity levels. In this study, time and frequency domain features were extracted from five accelerometers and a heart rate monitor. Tapia et al. concluded that the one statistical feature extracted from the heart rate monitor (number of heart beats above resting heart rate) led to an increase by only 1–2 percentage points in classification accuracy (from 94.6% for personalized models, 56.3% for LOSO trained models). In examining classifications performed using only the heart rate feature to find why it was so weak, Tapia et al. found many misclassifications happened at the beginning and end of more intense activities, due to changes in heart rate being delayed compared to changes in the physical activities' intensity.

Structural features examined in Lara et al. [2012] are seen in table 3.3. Of these, only first, second, and third degree polynomials were used in the final evaluation of the system. The paper also introduced two "transient features" (a term which has not been used in the context of feature extraction by other authors) for each vital data measurement to compensate for the delay effect seen in Tapia et al. [2007]: *slope* and *magnitude of change*, which are the sign of the direction of change and the amount of change in the measurement from the first to the last 20% of a time segment.

Table 3.3: Structural features examined in Lara et al. [2012]. Of these, only first, second, and third degree polynomials ended up being used in the final evaluation of the system.

Function	Equation	Parameters
Linear	$F(t) = mt + b$	$\{m, b\}$
Polynomial	$F(t) = a_0 + a_1 t + \cdots + a_{n-1} t^{n-1}$	$\{a_0, \ldots, a_{n-1}\}$
Exponential	$F(t) = a\|b\|^t + c$	$\{a, b, c\}$
Sinusoidal	$F(t) = a \times \sin(t + b) + c$	$\{a, b, c\}$

To assess the impact of the structural and transient features, the authors first selected the best combinations among eight machine learning techniques and three window lengths for two data sets: One with only accelerometer data features and one including the vital data features. In the comparison of these combinations, a 5 × 2 fold cross-validation for each combination was performed in five repeated runs. Activity-by-activity accuracies of the best-performing vital-data-included classifier and the two best-performing acceleration-data-only classifiers were compared in a 5 × 10 fold cross-validation in five repeated runs. As no LOSO evaluation is mentioned in the paper, we may assume data from a subject could present in both the training and test sets.

Vital data features led to an increase in recognition accuracy for three activities (*sitting*, *ascending*, and *running*) and a decrease in one (*descending*). From these results, Lara et al. conclude that the choice of including such structural features should depend on the activities recognized. However, the increase in average accuracy from the best-performing acceleration-only classifier to the best-performing vital-data-included classifier was only three percentage points, from 92.9% to 95.7%, and although the authors do not mention it, this is only marginally better than both the overall results and the increased accuracy achieved using statistical features extracted from one sensor in Tapia et al. [2007]. The increase could possibly have been accounted for simply by the increased amount of data due to the vital data sensor. The paper does not evaluate statistical features extracted from the same sensor.

According to Olszewski et al. [2001, p. 42], extracting the structural features used in Lara et al. [2012] is $\mathcal{O}(n^3)$, n being the number of data points within a segment. These features are therefore more computationally expensive than time and frequency domain features for the same data.

3.1.7 ML Problems Especially Relevant to HAR

Bulling et al. [2014, section 2.2] identify four challenges that are especially relevant in HAR when compared to other ML problems, which all relate somewhat to the collection of a well suited data set. These challenges are:

1. **Definition and diversity of physical activities:** By “definition”, Bulling et al. refer to an understanding of what defines the activities which the system is to recognize. This understanding will affect what separates the activities physically, thereby affecting what sensors should be used and how to collect a data set. They note that “human activity is highly complex and diverse and an activity can be performed in many different ways, depending on different contexts, and for a multitude of reasons”. The complexity of human activities may lead to both **intraclass variability**, which is activities of the same class showing little similarity, and **interclass similarity**, which is activities of different classes showing similarity. An illustration of this is shown in figure 3.5

2. **Class imbalance:** Certain activities may be disproportionately represented within a data set, “disproportionate” being understood as not having an equal amount of samples as the most represented activity. Simply gathering more data for some activities may not be feasible: They may place a physical strain on the subject (e.g. running is more exhausting than standing, sitting or walking) or just occur more infrequently

(e.g. drinking from a glass as opposed to sitting still). Section 2.1.4 explained ways of correcting class imbalance in an already collected data set. During the literature search for this thesis, examples of HAR systems using cost-sensitive learning and oversampling were found, but no examples of systems using undersampling.

3. **Ground truth annotation:** As explained in section 3.1.2, HAR systems based on supervised learning requires previously labeled time segments, which must be labeled by a human. The challenges related to these labels are twofold: Firstly, labeling this data requires a lot of human labor and is therefore expensive. Secondly, it is hard to acquire movement data in contexts representative of daily life, as a human annotator will often need information outside of the sensor data in order to interpret them correctly. Such additional information often comes in the form of video recordings, which may require subjects to perform the activities in a laboratory environment. As explained in item 1, the laboratory context may affect how activities are performed. The effects of using data gathered in a laboratory context for classification of acceleration data gathered outside of a laboratory will be further explained in an upcoming subsection.

4. **Data collection and experiment design:** Bulling et al. bring up two connected problems under this point: The lack of general-purpose data sets that can be used in different research efforts and the problem of designing an individual data collection experiment. The first problem affects the overall progress in the field, as it makes the results harder to reproduce. The second problem stems from the trade-offs which must be made when collecting a new data set: More data requires more sensors, which makes the system harder to use both in research and daily life.

The problems mentioned have an effect on both collection and use of data as well as the interpretation of results. As the design and execution of the data collection was not performed by the thesis' author, the following subsection will comment on a problem relevant to the interpretation of the achieved results: How systems trained on data gathered in a laboratory perform on data gathered outside of a laboratory.

3.1.8 Real Life Performance of Systems Trained on Data Gathered in a Laboratory

As explained in items 1 and 3 in section 3.1.7, activities may be performed differently depending on their context, an example of such a context being a laboratory. Most HAR systems for ambulatory activities are trained and tested exclusively on data collected in a laboratory. Only a few research papers have made an effort to validate the results of such systems outside of a laboratory setting. From now on, a *laboratory* data set will refer to a data set gathered in a laboratory, while a *non-laboratory* data set will refer to a data set gathered outside of a laboratory.

Foerster et al.

Foerster et al. [1999] studied ambulatory monitoring outside of a laboratory using four accelerometers and a microphone. Twenty-four participants wore accelerometers on the

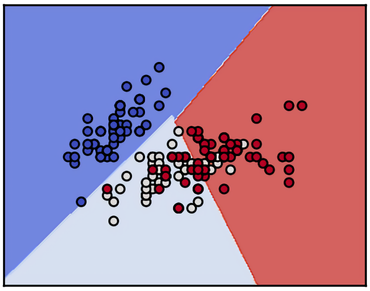


Figure 3.5: Interclass similarity and intraclass variability. A scatter plot of three classes according to two features of their samples. Blue samples show little interclass similarity to other classes as well as a low intraclass variability compared to the other classes. Red and white samples show much interclass similarity, as the values of their samples occupy the same regions in the feature space. Red samples also have a high intraclass variability, as they are spread more widely than those from the two other classes, but this variability is not so large that it would be a problem were it not for the white samples. Colored regions show the optimal division of the feature space given this training data. (Taken from from http://scikit-learn.org/stable/auto_examples/svm/plot_iris.html, ©2010–2016 Scikit-learn developers, BSD License)

wrist, thigh, lower leg, and chest and a microphone close to the throat. The activities recognized by the system were *standing*, *lying*, *walking*, *ascending stairs*, *descending stairs*, *cycling*, and three disjoint types of sitting (*working on a computer*, *talking* or just *sitting*). The researchers collected one minute of laboratory training data for each activity from each subject. The subjects then spent 50 minutes each outside the laboratory to provide a non-laboratory test set, performing whatever activity they wished under the supervision of a researcher who noted the start and end times of the activities. Afterwards, the subjects repeated the laboratory protocol to provide a laboratory test set. Classification was performed using a 1-nearest neighbor classification approach based on features extracted from 20 second segments.

The authors found that the system performed much better on the laboratory test set (95.8% accuracy) than the non-laboratory test set (67.7% accuracy). Short activities were found to be a large source of error. Eliminating all segments containing activities lasting shorter than 40 seconds from the test set led to the system achieving 81% accuracy. The authors also found that the two types of stair walking were often confused with regular walking and that the system had trouble differentiating the different types of sitting. By labeling all stair segments as *walking* and combining the different types of sitting into one *sitting* activity, effectively reducing the number of activities from 9 to 5, the system was able to achieve 95.5% accuracy on the non-laboratory test set, comparable to the accuracy on the laboratory test set. In short, using longer time segments and distinguishing fewer activities may be beneficial when training a system intended for non-laboratory use on laboratory data.

Bao and Intille

Bao and Intille [2004] performed a study of household activities (e.g. *vacuuming*, *read-*

ing, *working on a computer*) and ambulatory activities using five two-axis accelerometers (ankle, thigh, hip, upper arm, and wrist).

Data was collected using two different protocols. Neither of the protocols were performed under researcher supervision. Labels were based on start and end times of tasks, which the subjects wrote down before and after task execution.

- **Non-laboratory protocol:** Subjects perform activities outside a laboratory following a script. Tasks were designed to disguise the activities that the researchers were interested in (e.g. instead of telling the subjects to "work on a computer", they would be told to "use the web to find out what the world's largest city in terms of population is") and thus encourage natural movement patterns.

- **Laboratory protocol** Subjects perform activities following a script which stated more clearly what activity was to be performed, e.g. "walk without carrying any items on your back or in your hands". These sessions lasted about 75% as long as the non-laboratory sessions.

Two groups provided data for the collection:

- **Volunteers:** 20 subjects recruited from the researchers' campus, ranging from 17 to 48 years in age. Each volunteer contributed 1 run of the laboratory protocol and 1 run of the non-laboratory protocol.

- **Affiliates:** 3 subjects who were affiliates of the researchers, no age mentioned. Each affiliate contributed 5 runs of the laboratory protocol and 1 run of the non-laboratory protocol.

The authors performed a number of experiments with the collected data. Four of these experiments will now be presented. These were chosen because they were thought to best demonstrate how the non-laboratory data impacted classification. Their organization and naming are specific to this thesis.

In all of these experiments, decision trees learned with the C4.5 algorithm were used as classifiers. Features from the time and frequency domains were extracted from 7 s segments with 50% overlap for both training and testing. The classifiers were tested once on every subject in the group.

- **Experiment 1: Volunteers, personalized classifier.** Train a classifier on a volunteer subject's laboratory session (1 laboratory session). Test the classifier on the subject's non-laboratory data. Achieved a mean accuracy of 71.6% for all subjects.

- **Experiment 2: Volunteers, non-personalized classifier.** Train a classifier on both non-laboratory and laboratory sessions from all volunteers except one (19 laboratory sessions, 19 non-laboratory sessions). Test the classifier on the remaining subjects' non-laboratory data. Achieved a mean accuracy of 84.3% [Bao and Intille, 2004, figure 4].

- **Experiment 3: Affiliates, personalized classifier.** Train a classifier using all of an affiliate's laboratory sessions (5 laboratory sessions). Test the classifier on the subject's non-laboratory session. Achieved a mean accuracy of 77.3%.

- **Experiment 4: Affiliates, non-personalized classifier.** Train a classifier on one laboratory session from five randomly chosen subjects excluding one affiliate (5 laboratory sessions). Test on the excluded subject's non-laboratory session. Achieved a mean accuracy of 73.0% [Bao and Intille, 2004, figure 7].

These results may be interpreted in several ways. First of all, they seem to indicate that a larger amount of training data is beneficial to classification, as the the accuracy of classification is higher in experiments which use a larger amount of training data. Second of all, the difference between the results of experiments 3 and 4 indicate that personalized classifiers perform better than non-personalized classifiers.

Third and most importantly, the results indicate that non-laboratory data is beneficial to classification, as experiment 2 achieves better results than any other experiment, even those with personalized models. However, as this is the only experiment which makes use of non-laboratory data and also the experiment which uses the most training data, it is hard to quantify how much is due to more data and non-laboratory data.

Larsen and Vågeskar

The research project leading up to this thesis, Larsen and Vågeskar [2016], also investigated how a classifier trained on laboratory data performed on non-laboratory data.

The laboratory data set, which will be referred to as the Trondheim In-Laboratory (TIL) data set, had been collected for the master's thesis of Hessen and Tessem [2016]. This consisted of recordings of 23 subjects wearing two accelerometers (on the thigh and the upper back) performing nine different activities (*lying*, *sitting*, *standing*, *walking*, *running*, *stairs (ascending)*, *stairs (descending)*, *cycling*, and *bending*). Video of the activities was recorded by a chest-mounted camera. Subjects had to follow a script containing a sequence of activities lasting about 45 minutes in total for each subject.

The non-laboratory data set, which will be referred to as the Trondheim Free Living (TFL) data set, was collected from 11 subjects. Subjects were equipped with accelerometers on the lower and upper back and a chest-mounted camera, as in the TIL data collection, in addition to sensors on the lower back and one of the wrists. Subjects were fitted with the equipment at St. Olav's University Hospital and were free to do whatever they pleased in the area around the hospital for 3 hours, although subjects were encouraged to perform certain activities (such as *cycling*) and to try to minimize sitting time.

A convolutional neural network (CNN) with two convolutional layers and one fully connected neural network layer (developed for Hessen and Tessem [2016]'s master's thesis) was used to compare classification accuracy on the two different data sets, using the upper back and thigh sensors. Hessen and Tessem had originally found that this system could achieve a 97% accuracy when classifying the TIL data set. Larsen and Vågeskar found that overall accuracy decreased by nearly 20 percentage points (from 97% to 79%) when the system, using TIL as training data, was to classify the TFL data set [Larsen and Vågeskar, 2016, section 4.1]. The system exhibited very low recall and precision for *lying*, *bending*, and *cycling* outside of the laboratory. The latter was to be expected, as the body is exposed to more forces when cycling on a bicycle rather than an ergometer cycle. The system's performance on subjects who cycled a lot was consistently worse than its performance for other subjects. No clear explanation for the problems in recognizing *bending* or *lying* was found.

Several measures to mitigate the bad accuracy were tried out in other experiments, such as normalizing the input data and mixing data from TFL into the training set, but none of these measures had any significant impact on the overall accuracy of the system.

3.2 Accelerometer HAR for Stroke and Stroke-Like Disabilities

Section 2.6 explained stroke as a condition and stroke rehabilitation. This section will explain the motivation for using HAR for stroke patients, what challenges are especially important in this group, and some examples of HAR which are relevant to the experiments later in this thesis.

It is worth noting that the number of studies examining ambulatory HAR using accelerometers for stroke patients is considerably lower than the number of studies for healthy subjects. Cheung et al. [2011] performed a review of all English language HAR research using accelerometers published between 1980 and early 2010. The authors found 526 studies of activity monitoring in this group, of which 472 were excluded for reasons such as not using raw accelerometer data for activity monitoring[2] and being review articles. Out of the 54 remaining studies, 37 were concerned with healthy subjects and 17 with patients, only 2 of these concerned with stroke patients. Because of the small number of studies examining stroke patients, this section will also refer to examples of HAR for people with disabilities resembling those seen in chronic stroke patients.

3.2.1 Motivation

Capela et al. [2016] summarized the motivation for performing ambulatory HAR for stroke patients as follows: “For a clinician, reliable data about a patient’s activity is important, particularly information about the type, duration, and frequency of daily activities (i.e., standing, sitting, lying, walking, climbing stairs). This information can help therapists design rehabilitation programmes and monitor progress of patients outside of the hospital. [...] Mobility monitoring could provide large datasets with information about the mobility habits of people who have suffered a stroke, guiding future research in the field of healthcare and intervention.”

Outside of HAR, there are two main ways of assessing a patient’s physical activity: **self-reporting** by the patient and **observation** by a professional. Both of these methods have known weaknesses. Self-reporting by the patient relies on the patient’s perception, memory, and judgment. These cognitive capabilities are already imperfect in healthy subjects and can be made even worse by a stroke (see section 2.6.1). Self-reporting subjects can therefore over- and underestimate their abilities and activity durations, even intentionally misreporting data. Observance of the patient by a professional is less subjective than self-reporting, but is limited by the time and number of opportunities for assessment. Therefore, observation may not capture how the patient’s capabilities fluctuate throughout

[2] Instead of raw accelerometer data, the excluded studies used measurements *derived* from the data, such as step counts or “activity counts”, which indicate the intensity of the activity being performed.

a day and how they improve or deteriorate over longer periods of time. Thoroughly evaluated HAR systems can be a more reliable alternative to these assessment methods [Roy et al., 2009, p. 585].

3.2.2 Problems in Ambulatory Accelerometer HAR Targeting Stroke Patients

From the papers found in the literature search for this study, the main problem in ambulatory accelerometer HAR for stroke patients seems to be a difference in how activities are performed by stroke subjects when compared to healthy subjects (for an explanation of how diversity is a challenge to HAR in general, see item 1 in section 3.1.7). Systems trained on data collected from healthy subjects therefore consistently perform worse when applied to data collected from stroke patients.

This subsection will give examples of how this diversity has affected both the design and outcome of HAR research for stroke afflicted and similarly disabled subjects.

Lau et al.

Lau et al. [2009], the first paper on stroke patients examined by Cheung et al., presented a HAR system for recognizing different types of walking in hemiparetic stroke patients. Additionally, all of the subjects suffered from a dropped foot, a condition in which the patient is not able to adjust the forefoot's angle compared to the ankle. The paper is notable because of the large variation in walking within the group: Dropped foot patients often exhibit unusual movement patterns, like circumduction gait (moving the leg forward, extended, in a semi-circular motion around the hip instead of lifting it) and a high stepping pattern.

Seven subjects contributed data for five different types of walking: walking on level ground, up and down slopes, and up and down stairs. Two sensors with a sampling frequency of 240 Hz were attached to the back of the shoe heel and the top of the shank on the affected foot of each subject. These sensors contained a two-axis accelerometer as well as a single-axis gyroscope. Apparently, the authors segmented the sensor data stream by either automatically or manually identifying the start and end of each walking cycle, implying that the segments were of variable lengths. A support vector machine (SVM) was shown to be capable of achieving 82.9% accuracy in differentiating the different walking types.

Capela et al.

Capela et al. [2016] demonstrated the impact of using a system trained on data gathered from healthy subjects to classify data from stroke patients. Data was collected from 15 healthy subjects and 15 stroke patients in non-laboratory conditions and labeled with six activities: *standing*, *sitting*, *lying*, *walking*, *stairs*, and *small movement*. The only sensor used was a smart phone, worn on the front of the waist in a holster attached to the belt. This phone contained a three-axis accelerometer with a sampling frequency of 50 Hz. Time domain features of the acceleration signal, which had been separated into its gravity and linear acceleration components before extraction, were extracted from 1 second windows.

A decision tree classifier was hand-crafted using the researchers' knowledge of the extracted features' relevance to the given activities. The thresholds were adjusted to perform well on the data collected from healthy subjects. While the classifier was shown to be almost as accurate in differentiating mobile and immobile states in stroke patients as in healthy subjects, it was far less accurate when differentiating activities at a higher level of detail. Recognition for *walking*, *sitting*, *stairs*, and *standing* was considerably worse for stroke patients, attaining F_1-scores about 20–30% worse on these activities for the stroke subjects than for the healthy subjects. Inspecting the classifications, the authors concluded that many of these misclassifications could be attributed to the decision tree relying on information about the wearer's posture, which varied much more for stroke patients than healthy subjects.

Lonini et al.

Although not targeting stroke patients specifically, Lonini et al. [2016] examined HAR for patients with lower-body impairments. The group examined were patients wearing a knee-ankle-foot orthosis (KAFO). An orthosis is an assistive device which helps stabilize a weakened extremity, joint, or other body part, and the knee-ankle-foot prefix indicates that the leg, ankle, and foot are stabilized by this particular device. Several conditions could cause the need for such a device. Lonini et al. give the example of polio, but KAFOs can also be prescribed for stroke patients [Kakurai and Akai, 1996]. The goal of the paper was to investigate whether training data from the individual patient is needed to achieve adequate classification quality for an individual this group. The authors also examined the performance of a classifier trained on data from healthy subjects when classifying test data from the patient group.

Acceleration data was collected using a waist-mounted three-axis accelerometer with a sampling frequency of 30 Hz. Time and frequency domain features were extracted from 6 second windows with 75% overlap. Data was collected from 11 healthy individuals and 10 KAFO wearers. The patients' reasons for wearing a KAFO were not specified. The subjects had to contribute data in three sessions, each session about 35 minutes long and following an identical script. The script was aimed at collecting five different activities: *sitting*, *standing*, *walking*, *ascending stairs*, and *descending stairs*. The three sessions were performed on three different days to capture day-to-day variations in the subjects' movements.

RF classifiers were trained using three different approaches, named and described by the authors as follows:

1. **Global-healthy:** A 100-tree RF is trained on data acquired from the healthy subjects and evaluated on the KAFO subjects.

2. **Global-patients:** A 100-tree RF is trained on data acquired from all KAFO patients and tested on one KAFO patient in a LOSO fashion.

3. **Personal:** A 50-tree RF is trained on two of a subject's sessions and tested on the remaining session.

The approaches were compared in terms of median accuracy. The authors found that the global-healthy models were very inaccurate in classifying KAFO data (54.4% median accuracy). The global-patients and and personal models were almost equally accurate (81.0% and 84.2% median accuracy respectively). Confusion matrices for the three different approaches were also presented by the authors, and these showed that the only activities on which the global-patients approach performed significantly worse than the personal approach were the two types of stair walking. Not even the personal model was able to recognize more than 50% of the stair walking samples correctly. A follow-up experiment, in which data from all sessions were used to train personal models, revealed that the low accuracy in recognizing stair walking was largely due to variations in gait between sessions. The authors concluded that a population-specific model is sufficient to recognize activities within this group, and that this is probably true for other patient groups suffering from similar conditions.

3.2.3 HAR as a Diagnostics Tool

There are several examples of research on HAR systems which can be used to assess some attribute of a patient's level of physical ability other than pure ambulatory monitoring. This section will bring up some examples of such research.

Measuring Only Activities Which Indicate Functional Independence

Roy et al. [2009], which was the second paper on stroke patients found by Cheung et al. [2011], presented a system which was to recognize tasks in the functional independence measure (FIM). The FIM is a set of tasks which assess a patient's ability to live without assistance, e.g. cutting food, buttoning a shirt, and walking. The system was both to recognize whether a FIM task was performed and, in that case, what task was being performed. Otherwise it was to give no output.

Ten stroke subjects provided the system's data set. The setting and duration of the data collection sessions are not mentioned in the paper. Data for 11 activities in the FIM were collected, as well as for 10 activities *not* in the FIM, but which activate the same body parts and muscle groups (e.g. opening drawers, folding clothes, and standing). Eight single axis accelerometers were placed on the subjects, each of them complemented by a surface electromyographic (sEMG) sensor, which measures muscle activity. sEMGs can help distinguish intentional and unintentional movements. Both of these sensors had a very high sampling frequency, 1000 Hz, and time and frequency domain features were extracted from 4 second windows.

A combination of an artificial neural network (ANN) and a neuro-fuzzy inference system was used to make the classifier. Because not all windows would result in output from the system (the system was to give no output for non-FIM tasks), the authors did not use accuracy to measure the output quality. Instead, the system's sensitivity and specificity for the FIM activities were used as the system's quality metrics. These numbers were at 95% and 99.7% respectively, which means only 5% of the FIM task occurrences were wrongly labeled as non-FIM tasks and the system was only mistaken about what FIM task was being performed in 0.3% of the occurrences. The system's misclassification error, defined as the share of non-FIM tasks wrongly labeled as FIM tasks, was lower than 10%, although no exact number was given.

Estimating Walking Speed

As explained in section 2.6.3, walking speed is often used as an indicator of physical health in stroke patients. Systems which can estimate walking speed reliably could therefore be used as a tool to assess overall physical health. Dobkin et al. [2011] showed that accelerometers could be used to reliably estimate walking speed in hemiparetic stroke patients.

Dobkin et al. used three-axis accelerometers with sampling frequency 320 Hz worn on both ankles. Twelve stroke subjects provided three sessions of 50-foot indoor walking as training data, performed at slow, casual, and fast pace. Three outdoor walks with no particular instructions about pace were collected on a 67-foot stretch and used as a test set. Walking speeds in $^{m}/_{s}$, derived from stopwatch measurements of the time taken for each stretch, were used as labels for the data set. The walking speeds ranged from 0.4 to 1.2 $^{m}/_{s}$ in both the training and test sets.

Time and frequency domain features were extracted from the sensors. The authors did not mention any method for segmentation, and it is therefore reasonable to assume that these features were calculated from the entirety of each walking session. A naive Bayes classifier was trained on the indoor sessions and tested on the outdoor sessions. The system's estimates of outdoor walking speed were found to deviate by 6.7$\%$ on average from the stopwatch measurements. The system's output and the stopwatch measurements also showed a strong correlation, having a Pearson correlation coefficient of 0.98.

Gait-Based Diagnosis

Stroke is one of many diseases which can cause abnormal gait. Research into aspects of gait has a history which precedes the field of HAR, going back to at least the 1960s. One of the most important contributions to medical treatment from gait analysis has been the development of devices which can electrically stimulate muscles to make them contract at specific points in the gait cycle and thus assist patients with problems such as a dropped foot [Rueterbories et al., 2010].

A finding in gait research which is relevant to diagnostic HAR is that certain gait abnormalities can be characteristic of a given clinical population (a group of people sharing a disease or disability). This makes it possible to train HAR systems that can diagnose patients based on their gait, possibly detecting gait alterations which indicate the onset of disease before they become apparent to humans. A recent example of a system for detecting abnormal gait was presented by Mannini et al. [2016]. Their paper presented a system capable of accurately separating three groups of elderly patients based on accelerometer and gyroscope measurements.

Mannini et al. gathered data from 10 healthy elderly subjects, 15 post-stroke subjects, and 17 subjects with Huntington's disease (a genetic disorder which results in a gradual death of brain cells, one of its symptoms being abnormal gait). The subjects wore three inertial sensors (each containing both a three axis-accelerometer and a three-axis gyroscope with sampling frequency 128 Hz), placed on both shanks and on the lower back. Each subject contributed one minute of training data, which consisted of the subject walking back and forth a 12 meter stretch as many times as he or she was able to in this time period.

The authors made an interesting design choice when using the sensor data, which was to consider each side of the subject's body as a separate data stream. Each of the streams consisted of the data from one of the of the two shank sensors in combination with the lower back sensor. Their motivation for doing this was investigating whether there was a difference in how well the affected and unaffected sides in a stroke subject served in classification.

The streams were segmented so that each 12 meter walk was a separate time window. Time and frequency domain features were extracted from each such secgment in addition to probabilities from an hidden Markov model (HMM), which estimated the probability of the subject belonging to each of the three groups. Time domain, frequency domain and HMM features were used as inputs to an SVM, which would output a classification for each side of the body separately, resulting in two classifications for each 12 meter walk. Which population a subject belonged to was determined through a majority vote among all these classifications.

In a LOSO evaluation (leaving both sides of the evaluated subjects body out in training), the system was able to place 90% of the subjects in their correct populations. The authors also compared the differences in output from each of the stroke patient's sides, and found that the system made no errors when classifying stroke subjects only from their affected sides (two subjects were misdiagnosed as having Huntington's disease when the unaffected side was used for classification.

3.2.4 Optimal Accelerometer Placements for Recognizing Ambulatory Activities in Stroke Patients

Only one study found in the literature search for this thesis investigated the optimal number and placement of accelerometers for recognizing activities in stroke patients. This was the previously mentioned study by Roy et al. [2009]. Eight sensors yielded the best results, but the authors found that acceptable results (less than 10% misclassification) could be achieved using only four sensors. The optimal placements for four sensors were both upper arms, the preferred (dominant) forearm, and the unpreferred leg.

It should be noted that Roy et al.'s activity set included manipulative gestures (e.g. buttoning a shirt and cutting bread) in addition to ambulatory activities, which may have had an effect on which sensors were seen as optimal. In studies of healthy subjects, including manipulative gestures in the activity set has been shown to have an impact on the optimal set of sensors. For example, Bao and Intille [2004] (see section 3.1.8) found that a wrist and a thigh sensor was optimal for recognizing activities in healthy subjects given an activity set with manipulative gestures and ambulatory activities. Wrist sensors were found to be the worst sensor placements by Cleland et al. [2013] and Pannurat et al. [2017] (see section 3.1.5) using activity sets with exclusively ambulatory activities.

3.3 Semipopulation Approaches

Hong et al. [2016] presented a new take on HAR models, described by the authors as a "semipopulation-based approach". The article's research goal was to provide new users with personalized models based on a small amount of labeled data.

3.3.1 Training and Calibration

Usually, personalized models are created by gathering a large amount of data from the model's intended user and training a model using an ordinary ML technique on this training data. The idea of Hong et al.'s semipopulation classifiers is to use a smaller amount of data from each new individual to select the best among previously trained **sub-models**, which have been trained on data from other individuals. A sub-model is a classifier which is capable of recognizing one, and only one, activity. Figure 3.6 illustrates the semipopulation approach. There are two steps to the semipopulation approach:

1. **Training phase:** A pool of sub-models is built. This pool consists of one sub-model for each activity for each subject in the training set. Each such sub-model is trained on all the available feature windows from its subject to identify the presence or absence of only one activity.

2. **Calibration phase:** A small amount of feature windows from a user who is *not* in the sub-model pool is used to select the best sub-models available in the pool, i.e. no additional training occurs, only a selection among previously trained models. Let us call this set of feature windows the "calibration set". There are two calibration strategies: single-personalization (SP) and multi-personalization (MP). MP selects the best sub-model for each activity on an activity-by-activity basis, e.g. in figure 3.6, when calibrating a classifier for the new user (which we will call the "calibration subject"), MP selects the blue subject's sub-model for the first activity (e.g. *walking*), the green subject's sub-model for the next activity (e.g. *sitting*), and so on. SP selects the single user whose set of sub-models perform best for the calibration subject, and in the figure, this is the set of sub-models from the blue subject.

In the machine learning terminology from section 2.1.3, we could say that the calibration phase is a search of a very limited hypothesis space, because of a very strict language bias: The only hypotheses (sub-models) available are those that have already been found in the training phase. The implicit assumption is that a well-functioning hypothesis can be found within this very limited hypothesis space.

Hong et al. [2016] used a combination of a Bayesian network (BN) and an SVM to build each sub-model. When getting classifications from these models, the BNs would first be used to give an initial estimate of each activity's probability. This would be used to sort the activities by likelihood. After sorting the activities, system would go through the SVMs in order and see whether their probabilities were above a certain threshold, specific to each SVM. The first SVM to be above its threshold would define the output activity.

3.3.2 Hong et al.' Results

Hong et al. [2016] used a data set collected from 28 subjects wearing 6 three-axis accelerometers and a vital data sensor, which collected vital data in addition to acceleration. All sensors had a sampling frequency 10 Hz. The accelerometers were placed on both upper arms, in both trouser pockets, and on both ankles. The vital data sensor was placed on

[3] Reproduced from Hong et al. [2016] with the authors' permission.

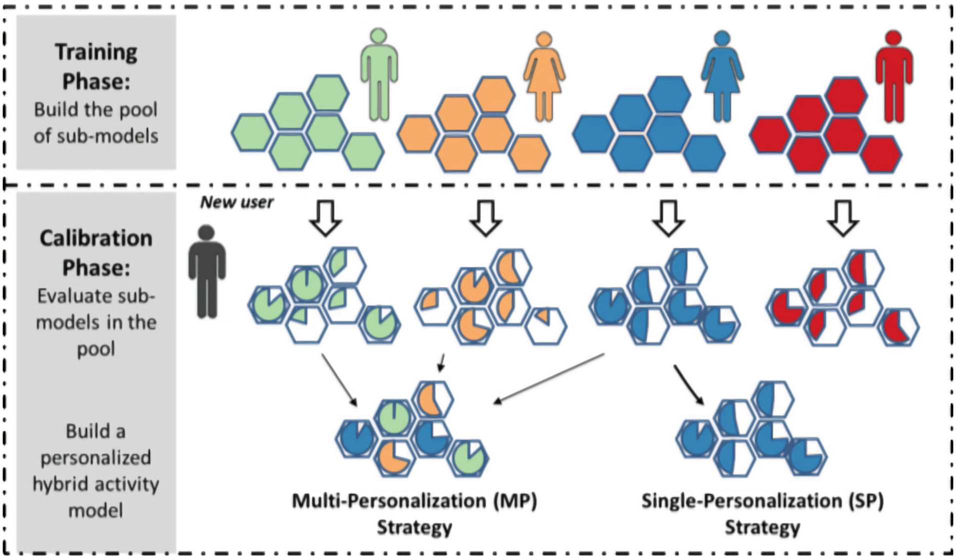

Training Phase:
Build the pool of sub-models
Calibration Phase:
Evaluate sub-models in the pool
Build a personalized hybrid activity model
New user
Multi-Personalization (MP) Strategy
Single-Personalization (SP) Strategy

Figure 3.6: Calibration phase in Hong et al.'s semipopulation-based approach[3]

the side of the chest. Collection sessions took place in a laboratory furnished to resemble a studio flat. Subjects performed two sessions in the laboratory, each following an identical script. The script instructed the subjects to perform different high-level activities (e.g. asking the subjects to wipe a desk instead of instructing them to bend down). Sessions took 1.5 hours to complete on average. Video recordings of the activities were labeled with seven activities: *sitting*, *walking (treadmill)*, *walking (indoor)*, *cycling (ergometer bike)*, *bending*, *lying*, and *falling*. Time domain features were extracted from 2 second windows with 50% overlap.

To calibrate a semipopulation model for a subject, the authors used one of the subject's sessions for calibration and the other session for testing. The authors found found that the MP strategy achieved 83.4% accuracy on average and that the SP strategy achieved 80.1% accuracy on average. Calibrating models based on other users' sub-models was found to be better than both training a set of sub-models one of the user's sessions and testing on the user's other session (77.3% accuracy) as well as using LOSO to train on the entire population and test on both of a subject's sessions (77.7% accuracy). The authors also experimented with reducing the amount of data used in calibration. They found that 22 minutes of calibration data was sufficient for a semipopulation MP model to attain the same accuracy as an individual model trained on 1.5 hours of training data.

Chapter 4

The Trondheim Chronic Stroke Data Set

This thesis introduces a newly collected data set, the Trondheim Chronic Stroke (TCS) data set. It was collected and annotated by Atle Kongsvold at The Faculty of Medicine and Health Sciences (MH) of The Norwegian University of Science and Technology (NTNU) and prepared for use by this thesis' author. The data set consists of timestamped accelerometer recordings annotated with ambulatory activities, i.e. activities that relate to moving one's body.

4.1 Equipment

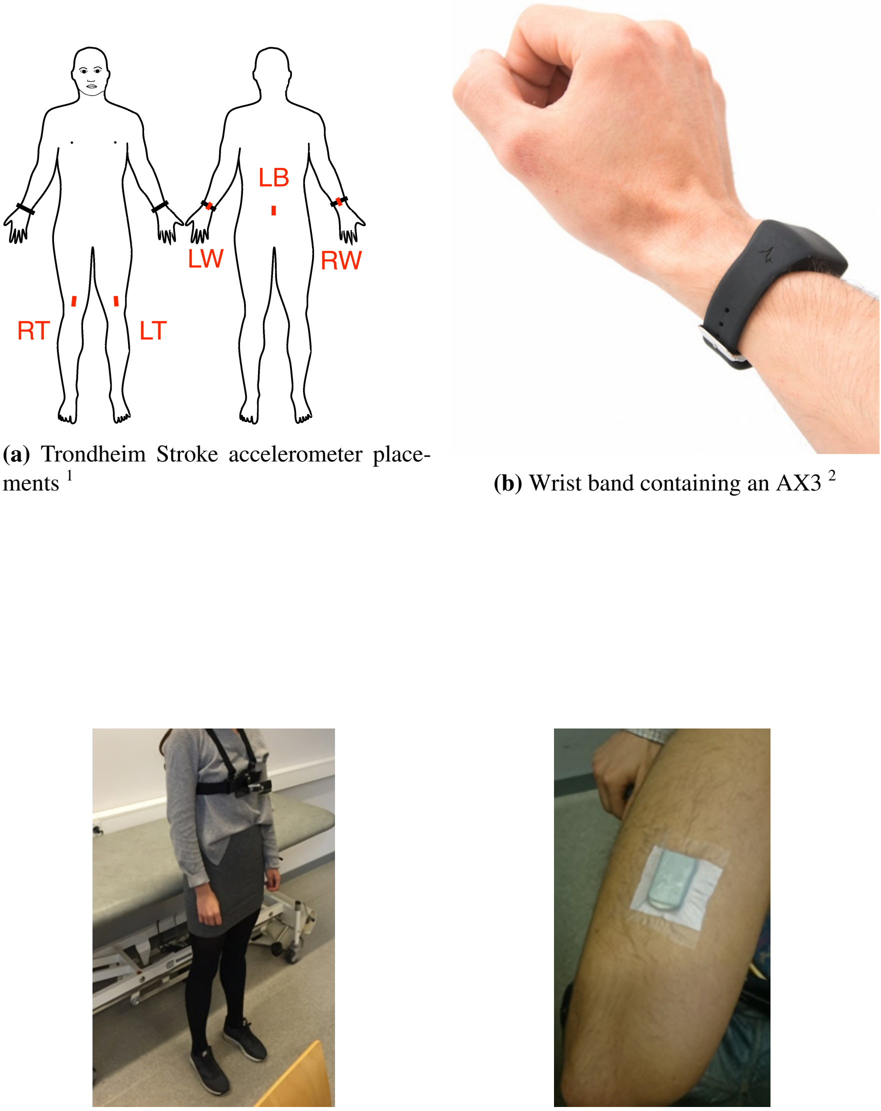
4.1.1 Sensors

Movement data was collected using five Axivity AX3 sensors (AX3s). These sensors contain a three axis accelerometer, sampling the acceleration for all three axes at 100 Hz. One sensor weighs 11 grams, and its dimensions are 23 mm × 32.5 mm × 7.6 mm.

Subjects wore the sensors in the following places: both wrists, both thighs (right above the knee), and on the lower back (center of L3, third lumbar vertebra), as seen in figure 4.1a. Wrist sensors were attached using wrist bands made by Axivity. For the remaining sensors, an area of the skin was covered in adhesive film of the Fixomull Stretch brand, and the sensors were attached to this area using double-sided tape. The area was then covered in a layer of protective film of the Opsite Fix brand. See figures 4.1b and 4.1d.

4.1.2 Camera

Video of subject activities was recorded in order to find the start and end times of the activities. The recordings were made by a chest mounted GoPro camera pointed towards



(a) Trondheim Stroke accelerometer placements [1]

(b) Wrist band containing an AX3 [2]

(c) Camera mounted on subject

(d) Accelerometer attached using adhesive film. Position above knee slightly higher than during collection.

Figure 4.1: Sensor and camera setup

the subject's feet, which provides a good view for distinguishing ambulatory activities. The setup is seen in figure 4.1c.

4.2 Collection Process

All data collection sessions were carried out in a room at The Department of Physical Medicine and Rehabilitation at St. Olav's University Hospital sufficiently large to perform the required activities in an unrestricted fashion. Subjects were equipped with sensors by a test leader and were accompanied by this test leader for the entire session duration. Data collection sessions followed a semi-structured protocol, the written version of which can be found in appendix A[3].

Subjects were required by the protocol to perform each of the following activities for at least five minutes in total: sitting, standing, moving (shuffling their feet while standing), walking (including stair climbing), biking (on a stationary bike), and lying down. Subjects could optionally jog or run freely, i.e. not on a treadmill. To provide data on bending and picking, subjects had to place objects found on the floor in a cabinet.

Originally, a sequence of three heel drops at the start of the video recording would provide a point of synchronization for the five sensors and the video annotations. As some subjects were not capable of performing these with sufficient strength, the heel drop procedure was left out of the recordings from subject S08 and on. To synchronize the sensors, the test leader would instead hold all sensors in one hand and clap both hands together, either before attaching them to or after detaching them from the subject. Start times for video recordings were written down to synchronize annotations and sensors.

4.2.1 Physical Tests

Two simple physical tests which give an indication of the subject's overall physical condition were also carried out: a timed 10 meter walk and a Timed Up and Go test (TUG)[4].

The 10 meter walk test simply required the subject to walk 10 meters as quickly as he or she was able to. Subjects were given a so-called flying start, meaning they were allowed to walk for a few meters before and after walking the 10 meter stretch, eliminating extra time spent accelerating. Bohannon [1997, table 4] found that men and women in the 50 to 59 age range typically achieve an average speed of 2.07 m/s and 2.01 m/s when walking 7.6 meters as fast as they can with a flying start. Using the arithmetic mean of these speeds (2.04 m/s), a healthy individual could be expected to walk 10 meters in 4.90 seconds.

The TUG test requires the subject to rise from a chair, walk three meters, turn around, walk back to the chair and sit down again. TUG is known for its test-retest reliability, and its results correlate to a large degree with other physical tests. For healthy subjects in the 60–69 age range (the lowest for which reference values are available), the average time to complete the test is 8.1 s [Bohannon, 2006, table 2].

[1] Modified from "Human Body Schemes" by Uwe Thormann (CC-BY-SA 3.0): https://commons.wikimedia.org/wiki/File:Human_body_schemes.png

[2] Photo courtesy of Axivity Limited (www.axivity.com).

[3] Details about the data collection differ from appendix A to this chapter. In such cases, this chapter is authoritative. All additions and differences have been supplied and confirmed by Atle Kongsvold.

[4] https://en.wikipedia.org/wiki/Timed_Up_and_Go_test

4.3 Subjects

Fifteen adults who have suffered one or more strokes make up the data set subjects. 10 were male and 5 female, 10 of them with an affected left side and 5 with an affected right side.

Table 4.1 summarizes the body measurements, physical test performance, and demographics for the subjects, whose individual data are found in full in appendix C. The subjects' scores on the 10 meter walk and TUG tests can also be seen as a scatter plot in figure 4.2

Table 4.1: Average body measurements and demographics for subjects in TCS data set

Feature	Age	Height	Weight	10 m walk	TUG
Average	55 years	174 cm	82 kg	11.36 s	16.26 s
St. dev.	11 years	8 cm	25 kg	6.74 s	7.07 s

4.4 Ground Truth Annotation

Subject videos were annotated using Michael Kipp's Anvil video annotation tool[5]. Fourteen different labels occur in the annotations. Appendix B presents the definitions used for labeling them based on the video from the chest mounted camera. Figure 4.3 shows their distribution in the data set.

1. Walking
2. Running
3. Shuffling
4. Stairs (ascending)
5. Stairs (descending)
6. Standing
7. Sitting
8. Lying
9. Transition
10. Bending
11. Picking
12. Undefined activity
13. Cycling (sitting)
14. Non-vigorous activity

[5] http://www.anvil-software.org/

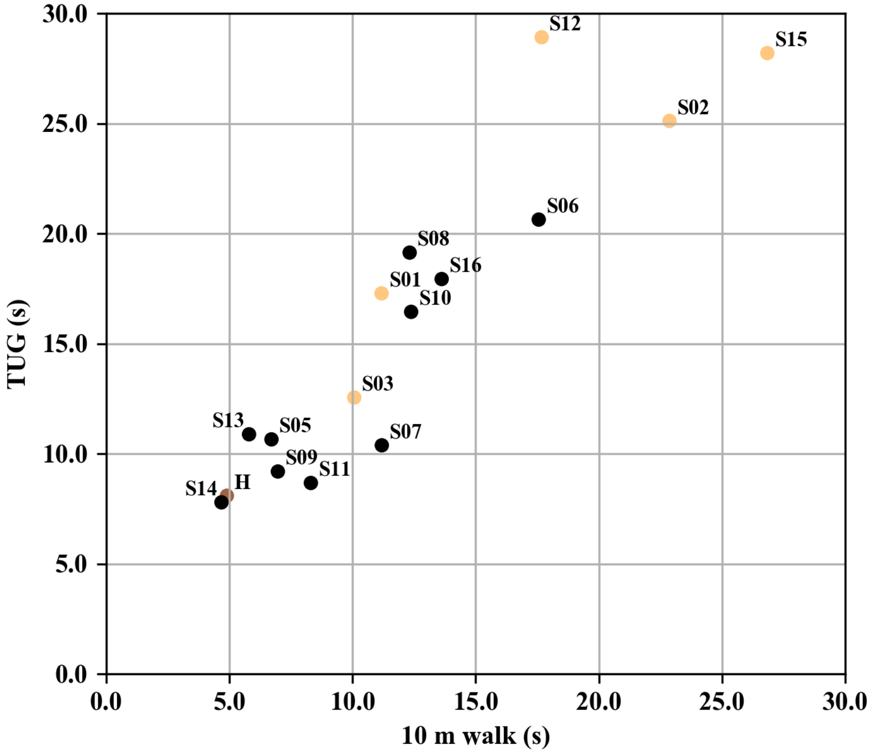


Figure 4.2: Scatter plot showing the relation between the subjects' scores on the 10 meter walk and TUG tests. Lighter colored dots are female. "H" marks the gender neutral reference values.

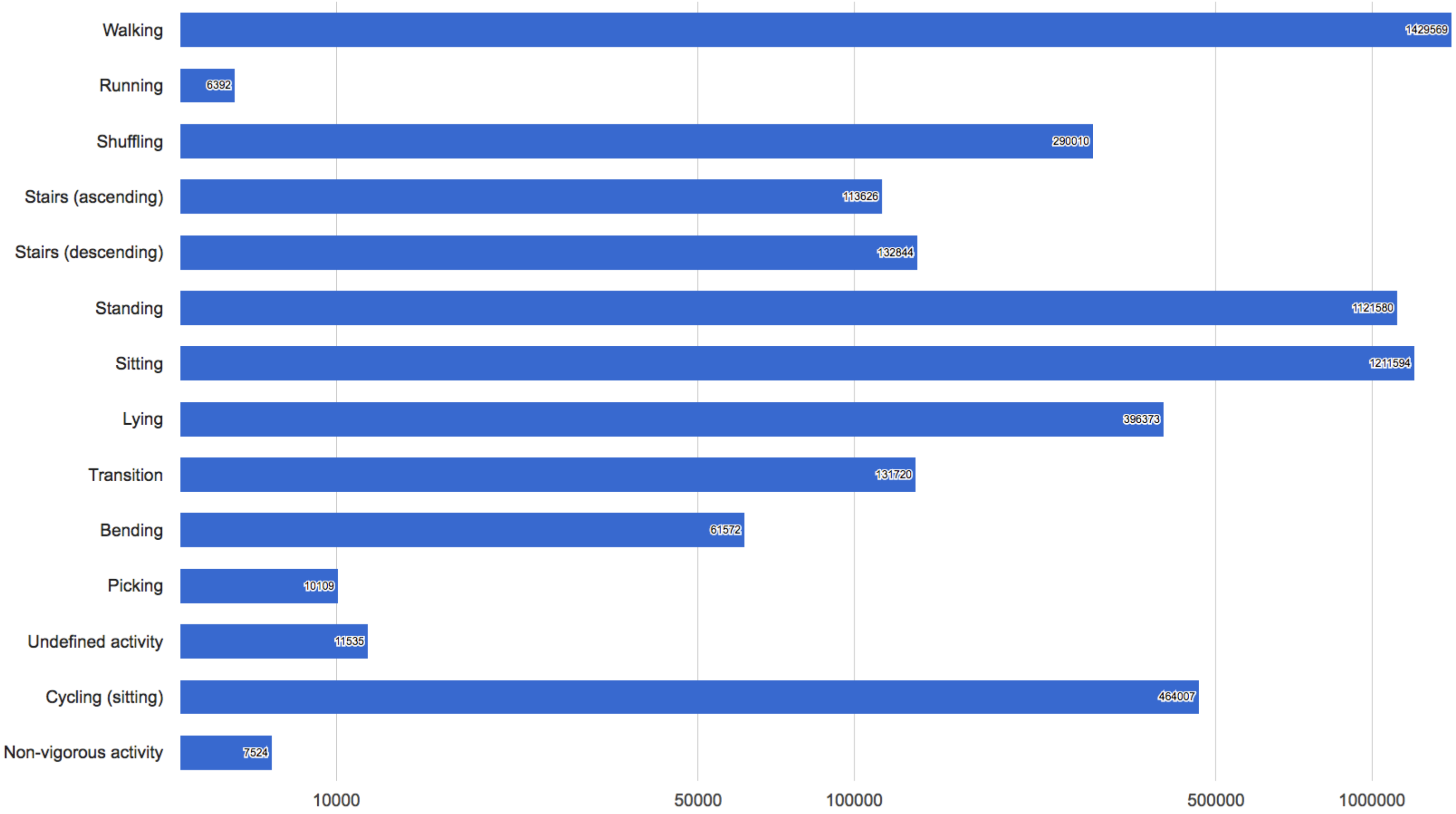


Figure 4.3: Logarithmic plot of the number of samples for each activity in the TCS data set. Sampled at 100 Hz, thus 100 samples is equal to 1 second.

Chapter 5

Methodology

This chapter will explain the methodology used for the experiments in the upcoming chapter. The explanation will follow the steps in the Activity Recognition Chain, which was presented in section 3.1.2.

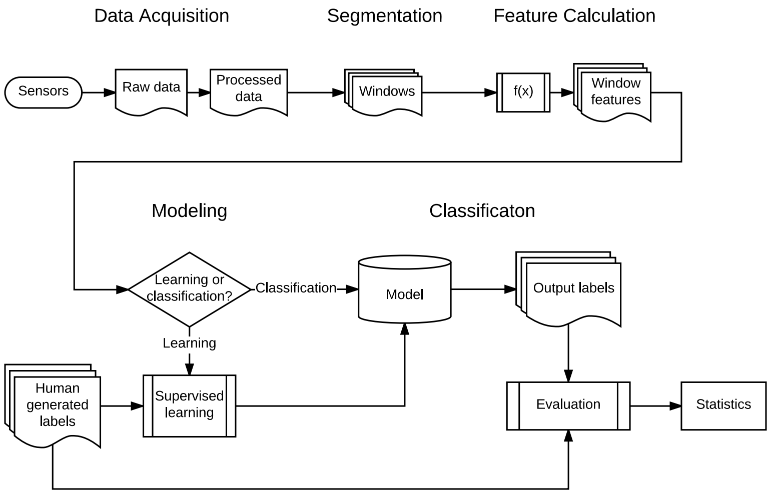


Figure 3.1: The Activity Recognition Chain (repeated from page 31)

5.1 Data Acquisition

Chapter 4 explained how the TCS data collection and ground truth annotation was performed. This section will explain how raw sensor data and video annotations were converted and synchronized in order to be used by the HAR system.

5.1.1 Sensor Synchronization

Axivity's OMConvert[1] and Timesync[2] software packages were used to convert and synchronize the sensor recordings. The OMConvert and Timesync tools perform one task each: OMConvert converts one AX3 raw data file (in the continuous wave accelerometry (CWA) format) to a WAV file. Timesync takes two WAV files as input and synchronizes them based on their magnitudes. The script's user decides upon one file whose values should remain unchanged (the master) and one file whose values can be re-sampled by the script (the slave). After synchronization, Timesync outputs the result as a seven column comma-separated values (CSV) file: three columns for each of the synchronized signals, with floating point numbers describing the acceleration along the X, Y, and Z axes in g_0s, and a column containing an absolute time stamp for each sample (taken from the master sensor). An illustration of this process is shown in figure 5.2.

Timesync is only capable of synchronizing two sensors at a time, but as only the slave sensor's values are changed, multiple sensors can be synchronized by selecting one sensor to be the overall master and synchronizing the remaining sensors with it in independent runs. When these independent runs are finished, all columns from the first two sensor synchronization are kept, and the remaining files' slave X, Y, and Z columns are appended to it. An illustration of this process can be seen in the top half of figure 5.3.

5.1.2 Synchronizing Video Annotations and Sensors

The video annotations output by the video annotation tool came in the form of a table. Each row in this table denoted a new occurrence of an activity. The columns specified such attributes as the activity's label and its start and end time relative to the start of the annotations.

Video annotations and sensor signals were synchronized using a Python script developed in conjunction with this thesis. The script requires that the user supplies the combined, synchronized sensor readings as a CSV as well as the video annotation file. The user also has to specify which row of the CSV at which the annotations start. The script then extracts the absolute time stamp from the annotation start row in the combined CSV. Subsequently, it adds this absolute time stamp to the relative time stamps of the annotations and labels all samples whose time stamp occurred between the start and end time of each annotation with the annotation's activity. After this has been done for all annotations in the video annotation table, the script deletes all rows of the combined CSV which have not been labeled. After deleting the unannotated rows, the script saves the label column and the remaining rows of each sensor's X, Y, and Z columns to their own separate CSV files. The result is six CSV files, all with an equal number of rows. Rows with equal indices correspond to sensor samples and labels from the same point in time. This process is illustrated in the bottom half of figure 5.3.

As said, the script requires that the user supplies the index in the sensor recordings at which the annotations were assumed to start. When using the script to synchronize the sensors and annotations of a subject for the first time, an assumed starting index was calculated by taking the difference in wall-clock seconds between the annotation start (noted by

[1] https://github.com/digitalinteraction/openmovement/tree/master/Software/AX3/omconvert

[2] https://openlab.ncl.ac.uk/gitlab/dan.jackson/timesync/tree/master

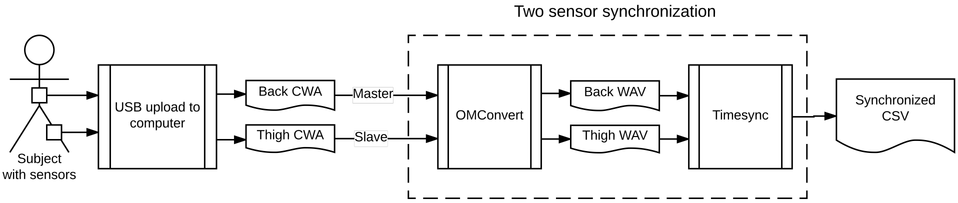


Figure 5.2: Synchronizing two sensors using OMConvert and Timesync

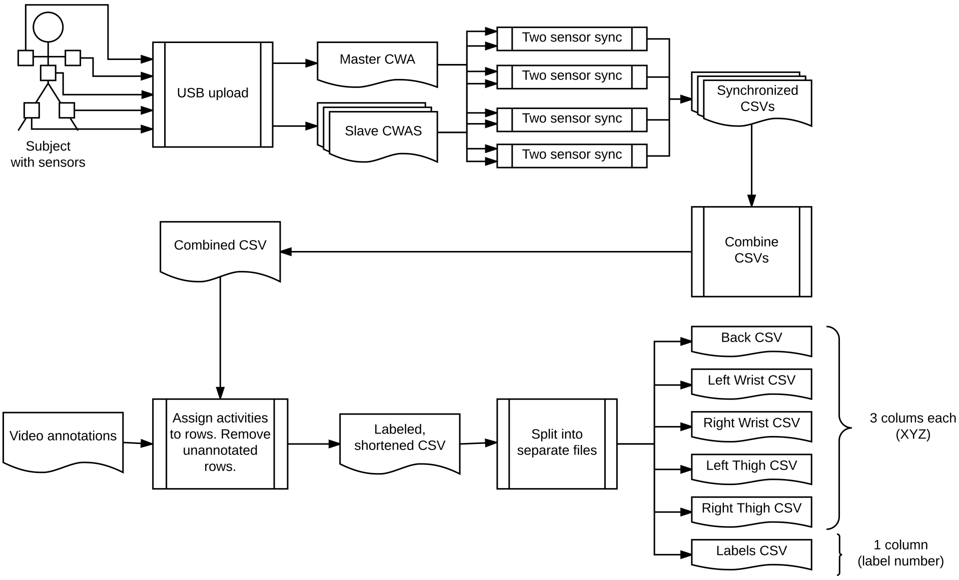
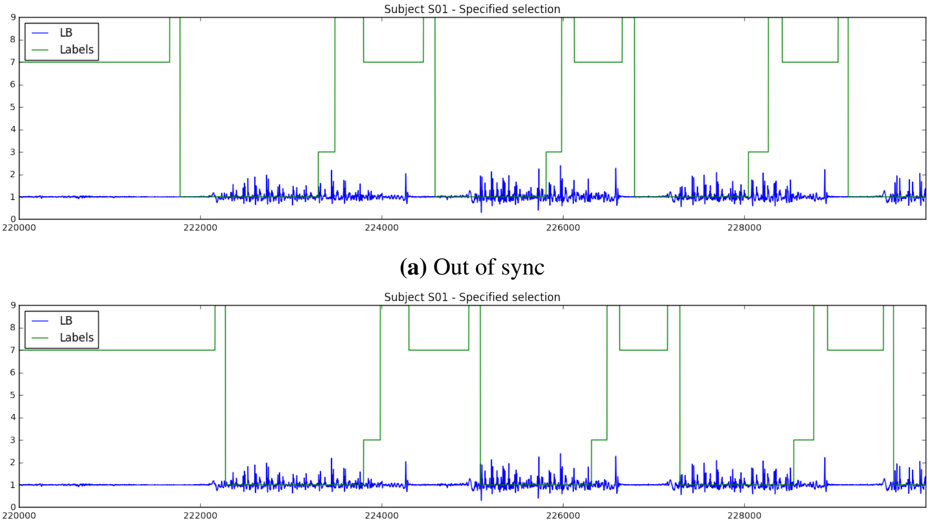


Figure 5.3: Synchronizing multiple sensors in addition to labels. The "Two sensor sync" procedure was presented in figure 5.2

the collector) and the sensors being turned multiplied by the sensors' sampling frequency (100 Hz). This value was off by several hundred samples in most cases. The annotation start index was adjusted by inspecting a time plot of the master sensor's magnitude and the output annotations and manually shifting the annotations until the plots lined up. Points of transitioning between high and low acceleration activities (e.g. walking and standing) were used as a guide for this process. Examples of synchronized and unsynchronized annotations during adjustment are shown in figure 5.4. When an appropriate starting index had been found, the synchronization script was run again using the correct starting index instead of the assumed starting index.



(a) Out of sync

(b) In sync

Figure 5.4: Synchronizing annotations with the master sensor. Label key: (1) walking, (3) shuffling, (7) sitting, and (9) transition.

5.1.3 Procedure and Adjustments

Synchronization was carried out using the procedures described in the two preceding subsections. The left thigh sensor was used as the master sensor for all subjects except one (S01), where only the lower back sensor would result in successful synchronization.

In the specialization project preceding this thesis, Timesync's outputs were in many cases found to be off by more than 10 samples [Larsen and Vågeskar, 2016]. Based on experiences from that project and from the preparation of TCS, Axivity has improved Timesync to the point where all but 1 of the 79 sensors used in TCS synchronize properly[3]. The outputs of the sensor synchronization and the sensor and video synchronization

[3] The problematic sensor was S02's right wrist sensor, which would not synchronize properly using any other sensor as a master. To compensate for an observed delay of 55 samples at video annotation start, its readings were shifted forward by 55 samples. An additional delay of 80 samples was observed at annotation end time,

processes were inspected using time plots, examining the synchronization at times close to the start and the end of the one hour recordings, and was found to be adequate for all subjects.

5.2 Segmentation

Window lengths of 1, 2, 3, 4, 6, 8, and 10 seconds were tested when designing the system. Windows of length 3 seconds, i.e. containing 300 samples, became the final choice, as this led to the best results. When extracting training data, an 80% overlap between windows was used. In testing, there was no overlap between the windows.

When extracting labels, each window was labeled with the most frequently occurring label within the window.

5.3 Feature Calculation

Features were extracted separately from each sensor using all the time domain features from table 3.1 and all the frequency domain features from table 3.2, which were selected because they were frequently occurring in the literature. The rest of this section will explain two choices that were made to cope with undefined calculation results and with cases where sensors had been wrongly attached.

5.3.1 Fixes for Undefined Results

In a rare number of cases (fewer than 20 windows in the entire data set), the results of calculating the correlation, spectral centroid, and spectral entropy features would be undefined or infinite. This would happen when the values along an axis were all zero or all had equal values within a window, possibly owing to some temporary error in the sensor or to some aspect of the Timesync synchronization script. The machine learning framework would not accept undefined or infinite values, and these values were therefore replaced by 0. As the number of cases where this happened was very small, the impact of this choice on the overall quality of the system was judged to be insignificant.

5.3.2 Coping with Wrongly Attached Sensors

Experiences with Wrongly Attached Sensors in the TFL Data Set

As explained in section 3.1.8, the CNN-based HAR system used in the project preceding this thesis performed dramatically worse for one subject in the TFL data set than the others. No explanation for this behavior was found before the project report had been delivered. As new subjects were added to TFL at the start of 2017, similar results were observed for some of the new subjects. To find an explanation for these bad-performing subjects, TFL was examined in cooperation with its collector, Atle Kongsvold. Inspections of the problematic subjects' raw sensor signals revealed that these subjects had all had been wearing

about 1 hour and 7 minutes later. This additional delay was not adjusted for.

at least one sensor upside down, leading to the signs of values along one or more of the sensor's axes being the opposite of expected. We found that excluding flipped-sensor subjects from the training set led to improved accuracy for the remaining subjects. Incorrectly attached sensors therefore seemed a likely explanation for the bad performance.

Planning for situations where sensors have been rotated is essential to the final quality of the system. If sensor data is collected following comparable procedures as were used to collect TFL or TCS, it is likely that similar mistakes will occur, as the AX3's exterior is nearly rotationally symmetric. As shown in the collection of TFL, even experienced professionals can make mistakes, as the sensor attachment procedure involves many steps, such as covering the sensor in water protective rubber, positioning and aligning it correctly with the body part, and attaching any adhesive film so that it is comfortable for the wearer. The subjects are also expected to be wearing the sensors outside of a controlled environment, and from related data collections, we have had reports of wearers who, for reasons such as itching or the adhesive film falling off, have re-attached sensors themselves, possibly with reversed axes.

Two approaches to reduce the impact of wrongly attached sensors were evaluated using the CNN-based HAR system. The first was to multiply all values along a reversed axis by -1. The second was to remove the sign of all sensor values from all sensor samples in the entire data set before they were used as inputs to the system, i.e. the absolute value of the acceleration would be used for learning and classification. Overall accuracy significantly improved using both techniques, owing mostly to fewer misclassifications in the subjects who had been found to wear flipped sensors. The first approach was slightly more accurate, but only by less than 1 percentage point overall compared to the second approach.

While there is a case to be made for only changing data which has found to be wrong, this approach requires manual inspection and modification. Approaches where values are automatically corrected, albeit with some loss of information, requires no intervention on the user's behalf. The system presented in this thesis could come to be used by people who may not necessarily have the technical knowledge needed to identify and modify the problematic axes, and even with the knowledge to do so, the task may require a lot of time if the sensor's position has changed at some point during a several day long collection. Consequently, choosing an approach which leads to a small reduction in accuracy, but a large reduction in the expected costs of labor, could therefore be worthwhile.

Removing the Sign of Calculated Features to Cope with Reversed Axes

Building upon the experience with wrongly attached sensors in TFL, a goal for the design of the feature extraction in this system was to make it so that sensors with one or more reversed axes would not have an impact on the system's output.

Using TFL and the CNN-based HAR system, removing the sign of all sensor values was an acceptable solution, as it did not require modifying the structure of the CNN itself. However, this changes signal's appearance considerably, and would lead to a significantly different result in many of the feature calculations used in this system. For example, the mean of a sine wave with amplitude 1 and frequency 1 Hz is 0 and its dominant frequency is 1 Hz, while the mean of the absolute value of the same signal would be positive (approximately 0.64) with a dominant frequency of 2 Hz.

To retain as much information about the original signal as possible while still coping with differences that could arise from reversed sensor axes, it was decided that only the final output from the feature calculations would have their signs removed, both during training and testing. This would only affect a few features, as most features in table 3.1 are already guaranteed to have a positive sign. The only features that will be affected by this operation are the mean, median, correlation and Hadamard product features of the time domain. As all a_j and f_j are positive and none of the calculations can introduce a negative sign[4], no features in table 3.2 are affected by this operation.

5.4 Removal and Relabeling of Activities

Windows labeled with certain activities were either removed or relabeled before the windows were used for modeling and classification. The removed activities (table 5.1) are the same as those removed by Hessen and Tessem [2016] and were removed for the same reasons. The relabeled activities (table 5.2) include picking, which was relabeled by Hessen and Tessem, but also stair walking. The exact definitions of the activities are listed in appendix B.

Table 5.1: Removed activities

Activity	**Justification**
Shuffling	Shuffling overlaps with two other activities: It is either a short walking bout or standing with some leg movements. This makes it difficult to recognize, and as it overlaps with two other activities, it is not a candidate for relabeling.
Transition	Transition is a movement between activities. Windows with this label show little similarity to each other.
Undefined activity	This activity contains activities which cannot be identified from the video or which occur before sensors and camera has been attached. Can contain a multitude of different activities, and recognizing it is therefore hard.
Non-vigorous activity	Activities which are recognizable, but do not classify according to the definitions are labeled as non-vigorous activity.

An example of an activity which the system did not recognize well, but which was neither removed nor relabeled, is running. There were two reasons for this: Few windows from other activities were misclassified as running, and the total number of windows for the activity was small. Including the activity did therefore not have a large impact on the overall quality of the system. Should more samples for the activity become available, the activity could come to be recognized better.

5.5 Choice of Random Forests for the Models

Experiments in chapter 6 will make use of two different modeling approaches. One is using ordinary RF classifiers, and the other is using an RF-based semipopulation approach inspired by Hong et al. [2016]. Semipopulation approaches were presented in section 3.3.

[4] The negative sign in the spectral entropy calculation is present to make the output of the calculation positive.

Table 5.2: Relabeled activities

Activity	Relabeled as	Justification
Ascending stairs	Walking	Ascending and descending stairs was often misclassified as walking. After bending, stairwalking was found to be the RF classifier's least accurate activity when compared to the CNN classifier in Hessen and Tessem [2016, table 6.5]. Recognizing this activity for stroke patients and other patients with gait impairment has previously been shown to be problematic by Lonini et al. [2016, figure 3] and Capela et al. [2016, table 5].
Descending stairs	Walking	Same as ascending stairs.
Picking	Bending	Hessen and Tessem found this to often be confused with bending because of high interclass similarity: Picking occurs when the subject places, touches, or picks up an object below knee height, which must occur in the middle of a bending activity. The activity is also the only manipulative gesture in the activity set, while all other activities are related to either locomotion or posture.

The specialization project which led up to this thesis made use of a CNN developed by Hessen and Tessem [2016]. One of the research goals for this thesis is finding the best combination of sensors for classification. Continuing with CNNs was ruled out because of time concerns, as a single round of training and testing takes 30 minutes using the resources available for this project. An experiment exploring e.g. the optimal sensor combination using LOSO evaluations could therefore take about a month at best, not including the work required to extend the CNN to work with several sensors, adjusting things like its structure and learning rate, and the risk of technical problems.

RF classifiers were considered a good alternative. First of all, it was the next best performing model type in Hessen and Tessem [2016, p. 51–52]'s evaluation of five different machine learning techniques for classifying the TIL data set, beaten only by CNNs[5]. Second, training RF-classifiers takes very little time compared to training a CNN: Training and testing on features extracted from the entire TCS data set with an RF classifier takes about 10 to 30 seconds (depending on the number of trees and sensors) using the CPUs on the previously mentioned served. Third, RF classifiers also have very few hyperparameters to tune when compared to other classification approaches [Lonini et al., 2016, p. 3266]. Fourth and finally, RF classifiers lent themselves very naturally to being used in a semipopulation approach.

The two following sections will describe the ordinary RF classifiers and the RF-based semipopulation classifiers used in this thesis.

5.6 Plain Random Forest Modeling and Classification

RF learning and classification was presented in section 2.2.4. The Scikit-learn machine learning framework's implementation of RF classifiers[6] was used. This uses the Gini

[5] The five techniques evaluated by Hessen and Tessem were J48 decision tree, SVM, CNN, RF, and naive Bayes classifiers

[6] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

index impurity measure. The non-default parameters used are as follows:

- **Number of estimators (i.e. trees): 50.** This number of trees was found to give a good balance between training time and accuracy. Higher numbers of classifiers were also tried out, e.g. 100 and 500 trees, which in the best case yielded accuracy scores about 1 percentage point higher than 50 tree classifiers. This was not enough to justify the increase in running time, proportional to the increase in the number of trees. Lonini et al. [2016], whose experiment setting and data set was quite similar to this thesis, also found that 50 trees was the best number of trees for a non-personalized model.
- **Class weight: 'balanced'.** The class weight parameter of Scikit-learn's RF classifier is used to control whether cost-sensitive learning is applied (see section 2.2.2). The 'balanced' option gives all classes equal weight. Equivalent to setting all M_c to an equal value in equation 2.6. As previously explained, equal class weights gives the same effect as oversampling the data set until an equal class distribution is achieved.

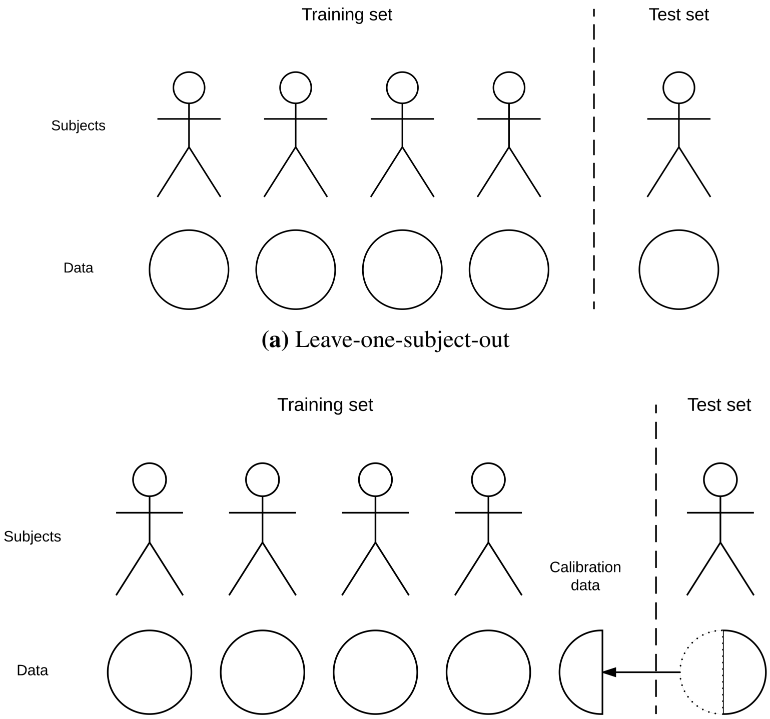
5.6.1 Leave-One-Subject-Out

The system is trained using data from all subjects except one and tested on the remaining subject. This is repeated for all subjects. The results achieved using LOSO will be considered as representative of how the system generalizes to new subjects that are not in the training set.

5.6.2 Mix-In

The semipopulation approaches require some data to calibrate a classifier for a new user, called the calibration set (the necessity of which will be explained in the upcoming section on semipopulation classifiers). To allow for a fair comparison between semipopulation and ordinary RF classifiers, experiments in which the results of semipopulation and plain RF classifiers are compared will present (in addition to the results of a LOSO classifier) the results of an RF classifier which has had access to the calibration set in addition to the ordinary LOSO set. These classifiers will be referred to as "mix-in" classifiers, as they have will have the same feature windows which are used for semipopulation calibration mixed into their ordinary LOSO training set. The difference between LOSO and mix-in is illustrated in figure 5.5.

The calibration data windows are taken from the test set (the reasons for this will be explained in the upcoming section), whose windows are extracted with no overlap. The data set they are added to, the training set, have been extracted with 80% overlap. A subject's test set data and training set data will consequently be five times smaller than and bigger than the other, respectively. To oversample the calibration data, so that each non-overlap calibration sample for an activity will be given the same weight as five training-set samples, each calibration data window's sample weight is multiplied by 5 before the data set is used for learning, i.e. it will have a sample weight 5 times higher than the other samples in its class.



(a) Leave-one-subject-out

(b) Mix-in, used for comparisons with semipopulation classifiers

Figure 5.5: Difference between the two approaches to training and testing plain RF classifiers

5.7 Semipopulation Modeling and Classification

A semipopulation approach inspired by Hong et al. [2016] will also be used in the experiments. In opposition to the ordinary RF classifier which was just explained, the aim of the semipopulation approach is not to make a general model whose intended user is any stroke patient, but to make personalized models whose intended users are one specific individual each.

The hope is not only that the semipopulation classifiers will perform well, but also that the users whose sub-models are selected by single-personalization (SP) and multi-personalization (MP) can give some indication of the calibration subject's health, as measured by a similarity in 10 meter walk and Timed Up and Go test (TUG) test results.

5.7.1 Sub-Model Design

This thesis' initial design for the sub-models was quite similar to Hong et al.'s. As the authors did not state the structure of their BNs in their paper, probability outputs from RF-classifiers were used in place of BN probabilities to initially sort the activities. Furthermore, each sub-model SVM's probability threshold was set to the threshold at which the classifier achieved the best accuracy in classifying its activity, as measured on its train-

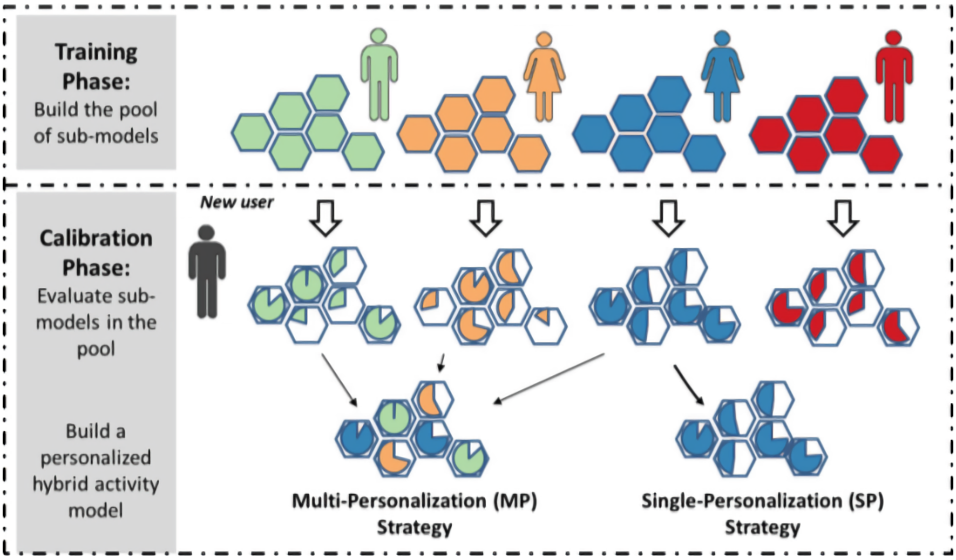

Training Phase:
Build the pool of sub-models
Calibration Phase:
Evaluate sub-models in the pool
Build a personalized hybrid activity model
New user
Multi-Personalization (MP) Strategy
Single-Personalization (SP) Strategy

Figure 3.6: Calibration in semipopulation approaches (repeated from page 53)

ing set. However, this approach was abandoned as the design never led to particularly good results and was additionally very computationally expensive when compared to what became the final design.

The final design of this thesis' sub-models was much simpler than both Hong et al.'s original design and this thesis' initial design, but led to much lower error rates than were achieved in the original paper (on a comparable activity set) and with the initial design. In the final design, each sub-model consisted only of an ordinary RF, trained to classify one activity. How these were trained will be explained in the upcoming subsection. When classifying, the probability output from the RF sub-models would be used to sort the activities by descending likelihood, and the most likely activity would be the output of the sub-model set.

Using SVM classifiers instead of RF classifiers was also tried during the design phase, but the results from using SVMs were never as good as those achieved using RFs.

5.7.2 Training: Building the Sub-Model Pool

The sub-model pool consists of sub-models from all subjects in the data set. Each sub-model has been trained on all available training data from one subject and is able to classify the presence and absence of one specific activity. The pool is constructed by repeating a single subject training procedure for all subjects in the data set.

The single subject training procedure requires a set of labeled feature windows. The procedure executes a single sub-model training procedure for all activities in the individual's training set. To construct a single sub-model, the system takes the subject's labeled feature windows and makes a copy of the labels. In this copy of the labels, all windows that are labeled with the specified activity are labeled as "true", and the remaining windows are

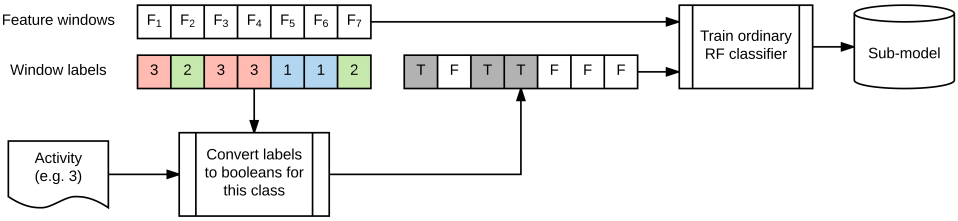


Figure 5.7: Sub-model training

labeled as “false”. Let us call these labels the boolean labels. An RF sub-model is then trained with the ordinary RF learning algorithm using the feature windows and the boolean labels corresponding to the windows. The output of the learning is an ensemble which will be used to estimate the probability of a certain activity occurring. An illustration of this procedure is seen in figure 5.7.

In this thesis’ system, 50 trees and balanced class weight (i.e. the true and false classes having an equal total weight) was used in learning, which are the same parameters as were used for creating the ordinary RF classifiers. Choosing balanced class weights was justified as the number of true samples would always be far smaller than the number of false samples. The number of trees, 50, was chosen because smaller numbers often led to two activities being equally probable, which would happen when an equal number of trees in their ensemble classified them as true.

5.7.3 Calibration: Selecting Sub-Models for a New User

In this thesis’ system, a new subject’s calibration set consists of a random sample of feature windows representing 20% of its collection session, with the remaining 80% of the windows being set aside for testing. The windows were selected so that the class distribution in the calibration set would be proportional to the overall class distribution for the subject.

The calibration feature windows were picked from the test set (no overlap), rather than the training set (80% overlap), to avoid using samples in the calibration phase that partially overlapped with samples which would be used in the test phase. Overlapping windows often have very similar feature values (two consecutive windows share 80% of their samples), and using such overlapping windows in both the calibration and test phases would not display the system’s ability to generalize, but its ability to memorize its training set. Using non-overlapping data from the same session is not a perfect solution, as neighboring windows may still contain very similar samples and thus have very similar features. Ideally, the calibration set and the test set should have been collected at different times to show the system’s ability to generalize. However, as the TCS data set does not include more than one session for any subject, a random subsample of non-overlapping windows is the best way to calibrate and test semipopulation models.

Accuracy on the calibration set is used to select best-performing sub-models for a user in both SP and MP. In SP, the system goes through the sub-model sets associated with each user and returns the sub-model set which led to the highest accuracy on the test set.

In MP, the system goes through all the sub-models associated with each activity, selecting the single sub-model which yields the highest accuracy on this user's calibration set.

Chapter 6

Experiments

This chapter presents the experiments which have been performed. Throughout it, the following abbreviations are used for the different sensor placements in the data sets:

- Lower back (lb)
- Left thigh (lt)
- Right thigh (rt)
- Left wrist (lw)
- Right wrist (rw)

There are some experiments in which sensors are matched between subjects not by whether they are on the same left or right limb, but by affected or unaffected limb. We will call these two matching schemes left-right matching and affected-matching respectively. The use of the back sensor is not changed in any way by these different matching schemes, as each subject only wears one (horizontally centered) back sensor. When subjects are matched by affected side, these alternative abbreviations for the wrist and thigh sensors are used:

- Affected thigh (at)
- Unaffected thigh (ut)
- Affected wrist (aw)
- Unaffected wrist (uw)

Note that the Trondheim Chronic Stroke (TCS) subject identified as S02 lacks data from a lower back (lb) sensor. Therefore, the subject is left out of training and testing whenever the sensor configuration involves the lower back sensors.

6.1 Training on Healthy Subjects and Testing with Stroke Patients

The first research goal of this thesis is to make a HAR system which performs accurate classification for subjects with motor impairments, i.e. stroke patients. This experiment will demonstrate the performance of the system when it is trained on data from *healthy subjects*, so that results from later experiments using stroke patient training data can be compared to it.

To establish how the system performs when trained on data from healthy subjects, we will run an experiment where the ordinary random forests (RF) classifier presented in section 5.6 will be trained on two separate data sets collected from healthy subjects: the Trondheim In-Laboratory (TIL) and Trondheim Free Living (TFL) data sets, which were presented in section 3.1.8.

TIL and TFL were selected because they both utilize AX3 sensors and are labeled with the same activities as the TCS data set. Both them are used, instead of just one, because each of them has a major advantage and disadvantage when they are to be used with TCS:

- **TFL has a lower back sensor, but was collected outside of a laboratory.** Having the same placement for the lower back sensor as TCS is an advantage to classification. Being collected in a different setting is a disadvantage, as subjects may perform activities differently. Additionally, TFL's cycling activity has been collected on an actual bicycle, which exposes the body to more accelerating forces than when cycling on an ergometer cycle, as in TCS.

- **TIL was collected inside a laboratory, but only has an *upper* back sensor.** The back sensor used to collect TIL was attached slightly higher up on the back than in TCS, which may change how and which forces are felt by the back accelerometer when compared to TCS. However, the TIL data set was collected in a laboratory, which may make the performance of the activities more similar to how they were performed in TCS. The *cycling* activity was performed on an ergometer cycle.

6.1.1 Setup

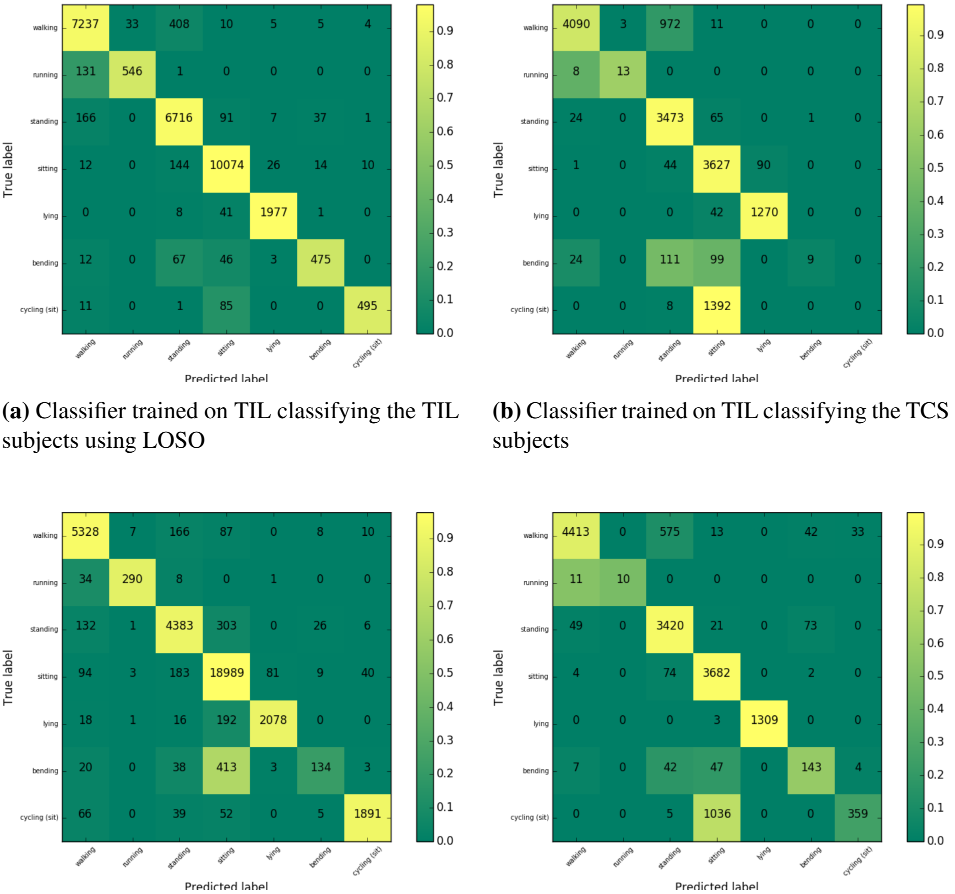
Ordinary RF classifiers will be trained separately on two data sets: the TIL and TFL data sets. They will then be tested on all TCS subjects, using the subjects' lower back and right thigh sensors, as well as on their own data set (using LOSO) for comparison. The experiment will be repeated ten times, and the statistics presented will be averages over these runs.

The right thigh from the stroke subjects was selected, as the TIL and TFL subjects also wore the thigh sensors on their right thighs. As subject S02 from TCS lacks a lower back sensor, it is excluded from testing.

6.1.2 Results and Discussion

Classifiers using TIL as a training set achieved 81.3% accuracy on average when classifying stroke patients, compared to 95.2% when classifying subjects from its own data set. For classifiers using TFL as a training set, the numbers were 86.5% when classifying stroke patients and 94.2% when classifying subjects from its own data set.

Confusion matrices for the two classifiers are seen in figure 6.1. As can be seen when comparing figures 6.1a and 6.1c, the TIL classifier performs better on its own training set than the TFL classifier does on its training set. But when comparing figures 6.1b and 6.1d, it is apparent that the TFL classifier is superior to the TIL classifier when classifying the TCS data set. The TIL classifier's lack of success with the stroke patients is probably due to the differing back sensor placement. As the TFL classifier is clearly the best of the two, we will only discuss its results from now on.



(a) Classifier trained on TIL classifying the TIL subjects using LOSO

(b) Classifier trained on TIL classifying the TCS subjects

(c) Classifier trained on TFL classifying the TFL subjects using LOSO

(d) Classifier trained on TFL classifying the TCS subjects

Figure 6.1: Confusion matrices for classifiers trained on healthy subjects classifying stroke patient data, taken from one run of the experiment.

Table 6.1 summarizes the performance of the TFL classifier on the TFL and TCS data sets using four different quality metrics. The classifier performs surprisingly well for most activities and achieves relatively high F_1 scores for *walking*, *standing*, *sitting*, and *lying*. The most problematic activities are *bending* and *cycling*. However, its F_1 and recall scores for *bending* is actually better when classifying stroke patients than healthy subjects. This could indicate that there is less variation in how stroke patients perform the bending activity when compared to healthy subjects, making this movement easier to generalize for stroke patients than for healthy subjects. Using stroke patient data in training will hopefully have a positive effect, especially on the lower scoring activities.

Table 6.1: Statistics for the TFL classifier

(a) Classifying TFL subjects using LOSO

Activity	F_1	Rec.	Prec.	Spec.
walking	0.944	0.951	0.938	0.988
running	0.913	0.871	0.958	1.000
standing	0.905	0.902	0.908	0.985
sitting	0.963	0.979	0.948	0.934
lying	0.938	0.915	0.962	0.997
bending	0.334	0.216	0.731	0.999
cycling	0.948	0.925	0.973	0.998

(b) Classifying stroke data

Activity	F_1	Rec.	Prec.	Spec.
walking	0.919	0.862	0.984	0.993
running	0.713	0.557	1.000	1.000
standing	0.890	0.962	0.828	0.940
sitting	0.860	0.980	0.766	0.903
lying	0.998	0.996	1.000	1.000
bending	0.587	0.608	0.568	0.993
cycling	0.393	0.253	0.883	0.997

Figure 6.2 shows a plot of the average accuracy of the TFL classifiers on each subject in the TCS data set (blue bars) along with a plot of the subject's walking speed on the 10 meter walk test (red bars). There seems to be some connection between the physical ability of the subject and how well the TFL classifier performs on it. Some exceptions exist: The classifier performs surprisingly well on S06 and S16, although their walking speeds are low. On the other hand, S01 has a surprisingly low accuracy given its walking speed. Later experiments will investigate whether this seeming connection between accuracy and physical ability persists when stroke patient data is used in training.

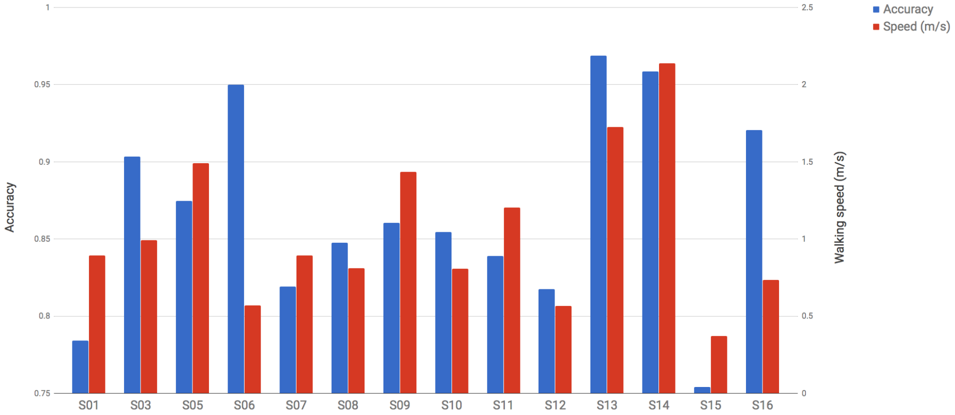


Figure 6.2: Subject by subject accuracies achieved by the TFL classifier plotted against walking speed.

6.2 Using Stroke Patient Data with Left-Right Matching

This experiment will test classifiers trained on stroke patient data with four different training approaches. The results of this experiment will be used to discuss four points of interest, which will now be presented.

Sensor placement

Five different sensors were used to collect the TCS data set. Knowing how the number and placement of these sensors affects classification quality is interesting for several reasons. First of all, finding a combination with a minimal amount of sensors which still achieves high quality is of interest to the data collectors, as a smaller number of sensors leads to less labor during collection and a lower probability of at least one sensor failing. Wearing fewer sensors will be more comfortable for the subject.

We may find that several sensor combinations are equally good or nearly as good for classification. If so, it will be possible to adapt the number and placement of sensors for subjects who for some reason (e.g. pain) cannot wear a sensor on one part of the body. For example, we could find that using the back sensor and one thigh sensor leads to the best results, but that two wrist sensors and a lower back sensor is nearly as accurate. Subjects who experience pain in their thighs could then wear sensors on the wrists instead of the thighs.

Stroke Patient Training Data

The previous experiment evaluated the performance of classifiers trained on healthy subjects and tested on stroke patients and found that there was some connection between a stroke patient's physical ability and the classifier's success in classifying the subject's activities. In this experiment's discussion, we will examine how classifiers trained on stroke patients perform when tested on stroke patients, both overall and on individual activities. We will also see whether the seeming connection between physical ability and accuracy persists.

Personalized Models

This experiment will also compare personalized models and non-personalized models. The personalized models will get access to 20% of the subject's data and use this to make a personalized model. Making a personalized model for *every* new subject is not a good solution if the system is ever to be widely used: As 12 000 people suffer a stroke each year in Norway (see section 2.6.2), the costs associated with gathering even a little amount of human-annotated data for just a fraction of them would be too high to justify. However, making personalized models may be necessary for *some* subjects, i.e. for subjects where a non-personalized model performs badly.

If we find that personalized models are better than non-personalized models for the same amount of sensors, using personalized models to reduce the number of required sensors could also be an option. For this solution to be practical, we must find a way to label training data automatically. This could be possible if the classifications output by a non-personalized classifier using a high number of sensors are found to be accurate enough to serve as training data. For example, we could have the subject wear five sensors for a short time, e.g. an hour or a day, and use the collected data and the predicted labels for this short period to train a personalized model, so that the subject would later have to wear only one or two sensors.

Semipopulation Classifiers as an Alternative to Ordinary Personalized Models

With regards to personalized models, another thing we would like to know is whether the semipopulation approaches, multi-personalization (MP) and single-personalization (SP), are good alternatives to personalized RF models. If MP and SP classifiers provide better classification quality using the same amount of data, using semipopulation classifiers instead of regular RF classifiers for personalization would be preferred. Also, an upcoming experiment will investigate whether semipopulation classifiers can be used to find similarities in physical ability between subjects. Knowing which sensor combinations lead to well-performing semipopulation classifiers will be useful to the interpretation of this experiment's results.

6.2.1 Setup

All possible sensor combinations, using any number of sensors between 1 and 5, will be tested using four types of classifiers:

1. **Plain random forests classifier, abbreviated "LOSO".** Performs a leave-one-subject-out evaluation for all subjects in the training set. This is the only non-personalized classifier, as it has has not been exposed to any data from the test subject before testing. It is therefore the only classifier whose results can be considered representative of the system's performance for subjects that the system has not had access to any training data from.

   Some samples from the test subject will be used in the Mix-in classifier's training set and in the semipopulation classifiers' calibration set. These samples will be removed from the LOSO classifier's test set, so that its test set is identical to that of the four other approaches.

2. **Mix-in random forests classifier, abbreviated "Mix-in".** Performs a "faux" leave-one-subject-out evaluation. The samples which are to be used in the semipopulation classifiers' calibration set are mixed in with the training data from the other subjects (explained in detail in section 5.6.2). Its purpose is to provide a baseline for the success of the semipopulation classifiers: If the semipopulation approaches are to be considered better than regular RF classifiers, they should perform better than this classifier.

   As this classifier has had access to data from the test subject, its performance is *not* representative of the system's performance for unknown subjects.

3. **Semipopulation multi-personalization classifier, abbreviated MP**: Classifier calibrated according to the MP strategy, explained in section 5.7.3. Performed for all subjects.

4. **Semipopulation single-personalization classifier, abbreviated SP**: Classifier calibrated according to the SP strategy, explained in section 5.7.3. Performed for all subjects.

Let us use the phrase *calibration set* to refer to the set of samples from the test subject which are either to be deleted from the LOSO classifier's test set, mixed into the Mix-in classifier's training set, or used for calibration by the MP and SP classifiers. This set will consist of 20% of the subject's test set samples, and the remaining samples will be used for testing. These samples are selected at random by a procedure which ensures that the class distribution in a subject's calibration set is the same as in its entire test set.

The experiment will be run ten times, and the results presented will be the average over these runs. All classifiers in each run will use an identical, randomly selected seed value as input to the calibration set selection procedure, ensuring that exactly the same samples take part in each subject's calibration set for all classifiers.

6.2.2 Results and Discussion

Table 6.2 shows the accuracy scores achieved for the different sensor combinations using the four classification approaches. Figure 6.3 shows confusion matrices from example runs of the the best performing LOSO classifiers (in addition to the single-sensor classifier for the lower back). Figure 6.4 shows equivalent confusion matrices for the Mix-in classifiers. Tables 6.3 and 6.4 show statistics for the corresponding classifiers. Figure 6.5 shows a plot of the accuracies achieved for each subject using the two sensors in the previous experiment and this experiment compared with walking speed. Figure 6.6 shows how accuracy changes for each subject as more sensors are being used.

The points presented in the experiment's introduction will be addressed one by one.

Sensor Placement

The LOSO classifier's results will be discussed here. When sensor placement is relevant to the discussion of the personalized classifiers, it will be mentioned in the discussion of these.

The LOSO classifier's results are in keeping with the findings about sensor placement for healthy subjects presented in section 3.1.5: Thighs and wrists are respectively most and least beneficial to classification, as seen in table 6.2a. This confirms Pannurat et al. [2017]'s results. The optimal sensors placements differ from those found by Roy et al. [2009] (explained in section 3.2.4). This is in in harmony with similar results for healthy subjects: Roy et al. found that a sensor near the wrist was essential to classifying their data set, which contained manipulative gestures in addition to ambulatory activities. This thesis' system recognizes ambulatory activities exclusively. Studies with healthy subjects have found that the wrist is beneficial when classifying manipulative gestures, but unnecessary when classifying only ambulatory activities.

Increasing the number of sensors beyond two has only a marginal impact on accuracy, and using one back and one thigh sensor outperforms all other two-sensor combinations. This is consistent with the findings of Cleland et al. [2013]. No greater difference in the

Table 6.2: Accuracy scores using 1 to 5 sensors and three different approaches. Sensor combinations are sorted in descending order by their LOSO classification accuracy.

(a) 1 sensor

lb	lt	rt	lw	rw	LOSO	Mix-in	MP	SP
	x				85.79	93.16	92.71	88.03
		x			84.95	91.87	92.88	88.77
x					74.17	84.40	78.85	75.81
				x	60.10	72.62	66.33	62.34
			x		55.11	70.74	64.67	58.28

(b) 2 sensors

lb	lt	rt	lw	rw	LOSO	Mix-in	MP	SP
x		x			93.17	96.30	95.13	93.71
x	x				93.07	96.24	94.50	92.86
	x			x	87.91	94.01	92.62	88.13
	x	x			86.62	94.85	93.37	90.78
	x		x		85.09	94.41	92.22	86.77
		x		x	85.07	93.21	92.05	88.40
		x	x		85.00	93.31	91.61	87.49
x				x	77.49	88.56	80.51	76.91
x			x		74.97	87.19	77.33	74.64
			x	x	65.42	79.71	68.70	60.98

(c) 3 sensors

lb	lt	rt	lw	rw	LOSO	Mix-in	MP	SP
x	x	x			94.28	96.69	95.10	93.89
x		x		x	93.57	96.72	94.55	92.97
x	x			x	93.44	96.72	94.31	93.30
x	x		x		92.62	96.58	93.81	92.39
x		x	x		92.54	96.58	94.37	92.78
	x	x		x	87.45	95.13	93.70	90.40
	x	x	x		87.19	95.52	93.23	90.01
	x		x	x	86.62	94.59	91.75	86.17
		x	x	x	85.46	93.78	91.23	87.44
x			x	x	79.70	89.48	78.98	74.77

(d) 4 sensors

lb	lt	rt	lw	rw	LOSO	Mix-in	MP	SP
x	x	x		x	94.59	97.04	95.21	93.54
x	x	x	x		93.91	96.99	94.91	93.67
x		x	x	x	93.02	96.82	93.80	92.15
x	x		x	x	92.96	96.83	93.51	92.47
	x	x	x	x	87.33	95.52	93.28	89.51

(e) 5 sensors

lb	lt	rt	lw	rw	LOSO	Mix-in	MP	SP
x	x	x	x	x	94.27	97.20	95.00	93.51

overall classification results are seen as the number of sensors increases in figures 6.3c through 6.3f.

Bending is the activity for which more sensors makes the largest statistical difference. Examining the statistics in tables 6.3c through 6.3f, *bending*'s F_1-score is 0.577 using two sensors, compared to 0.730 using five sensors. This is mostly due to fewer misclassifications of other activities (higher precision), and less due to more correct classifications of *bending* (higher recall). Figures 6.3c through 6.3f show that *walking*, *standing*, and *sitting* had most samples misclassified as *bending*. Less than 1% of the samples within these activities were misclassified as *bending*. The actual consequences of this quantitatively large improvement can therefore be considered insignificant.

The thigh and lower back sensors seem to complement each other in which activities they serve to distinguish. This can be seen when comparing figures 6.3a and 6.3b as well as tables 6.3a and 6.3b. For example, the back sensor leads to problems distinguishing *sitting* and *standing*. This is explained by the back having the same angle in relation to the force of gravity when standing and sitting. Angle compared to gravity also explains why the thigh sensor leads to the classifier confusing nearly all *lying* samples with *sitting*. This is in keeping with Veltink et al. [1996] (explained in section 3.1.4), who found that having sensors on the torso and one leg was necessary to distinguish *standing*, *sitting*, and *lying*. Additionally, the left thigh is to be sufficient to recognize *cycling*, and the lower back sensor is sufficient to recognize *bending*.

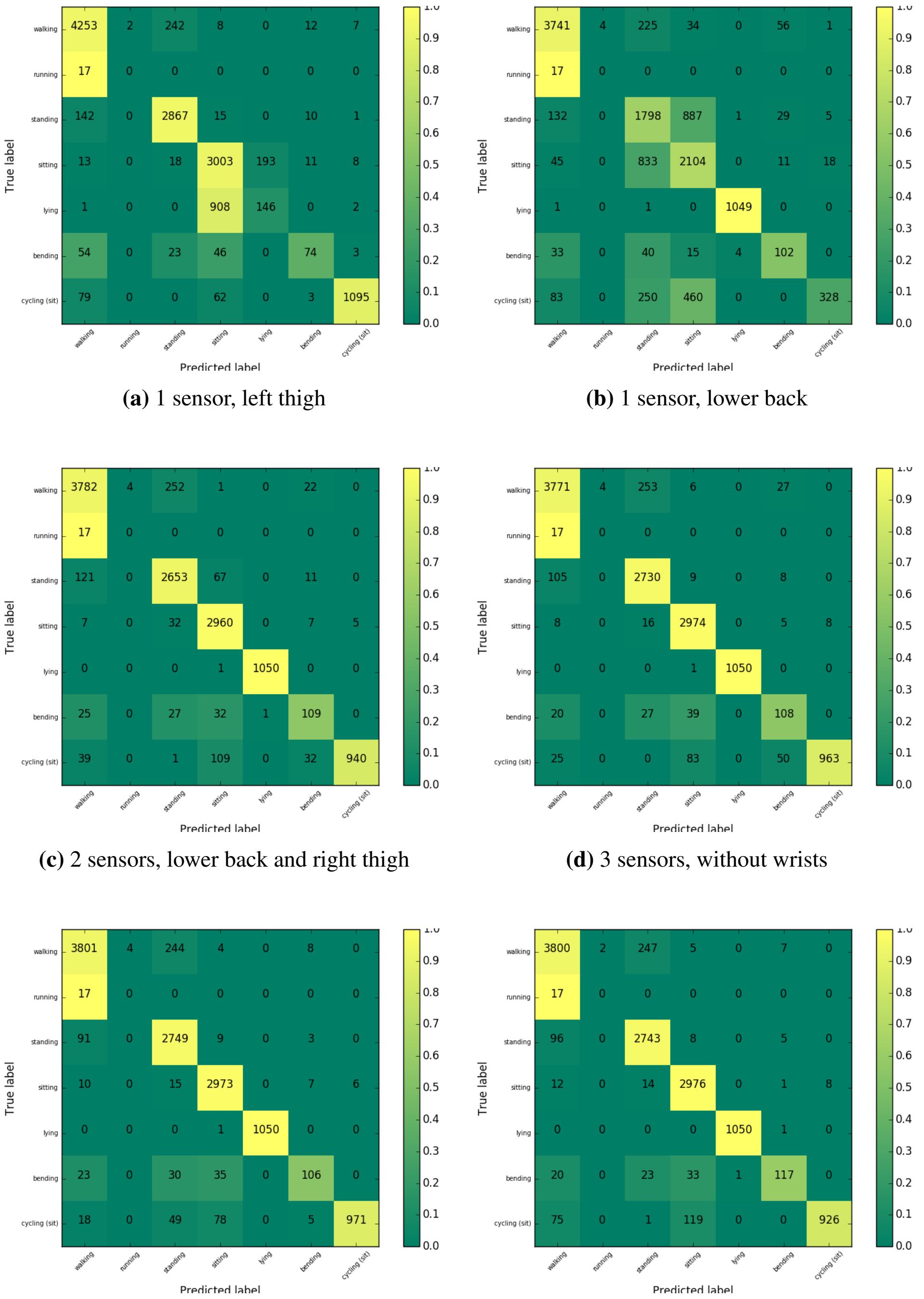
Figure 6.6 shows the increase in best accuracy as more sensors are introduced. Two sensors are sufficient to achieve more than 90% accuracy for most subjects, but accuracy is noticeably higher for subjects S05, S08, and S12 when three sensors are used. Subject S01 is the only subject for which using only one sensor was most successful. This indicates that something is different about the values extracted from S01's lower back sensor. The subject could possibly have a different posture than the others, or its lower back sensor could have been wrongly attached.

From the results, we can recommend using either two or three sensors, attached to the back and one or two thighs. The choice is a trade-off between cost and accuracy: An additional sensor increases sensor costs, the risk of one sensor failing, and storage requirements by 50%, but leads to slightly higher accuracy.

Adapting the sensor combination for users who need it is possible: The impact of removing one thigh sensor from the optimal three-sensor configuration is not very large (compare the top row of table 6.2c with top two rows of table 6.2b), and the same can be said for using the right wrist sensor instead of one of the thigh sensors (compare the top three rows of table 6.2c). Using these sensor configurations is therefore an alternative for subjects who can not wear sensors on the thighs. The lower back sensor is part of all sensor configurations which achieve more than 90% accuracy. Replacing it is therefore not an option.

Stroke Patient Training Data

Training data from stroke patients is significantly better than training data from healthy subjects when classifying the TCS data set. The overall accuracy on TCS when using training data from healthy subjects in the previous experiment was 86.5%. When trained on the same sensors in TCS, the classifier achieved 93.2% accuracy overall.



	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	4253	2	242	8	0	12	7
running	17	0	0	0	0	0	0
standing	142	0	2867	15	0	10	1
sitting	13	0	18	3003	193	11	8
lying	1	0	0	908	146	0	2
bending	54	0	23	46	0	74	3
cycling (sit)	79	0	0	62	0	3	1095

(a) 1 sensor, left thigh

	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	3741	4	225	34	0	56	1
running	17	0	0	0	0	0	0
standing	132	0	1798	887	1	29	5
sitting	45	0	833	2104	0	11	18
lying	1	0	1	0	1049	0	0
bending	33	0	40	15	4	102	0
cycling (sit)	83	0	250	460	0	0	328

(b) 1 sensor, lower back

	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	3782	4	252	1	0	22	0
running	17	0	0	0	0	0	0
standing	121	0	2653	67	0	11	0
sitting	7	0	32	2960	0	7	5
lying	0	0	0	1	1050	0	0
bending	25	0	27	32	1	109	0
cycling (sit)	39	0	1	109	0	32	940

(c) 2 sensors, lower back and right thigh

	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	3771	4	253	6	0	27	0
running	17	0	0	0	0	0	0
standing	105	0	2730	9	0	8	0
sitting	8	0	16	2974	0	5	8
lying	0	0	0	1	1050	0	0
bending	20	0	27	39	0	108	0
cycling (sit)	25	0	0	83	0	50	963

(d) 3 sensors, without wrists

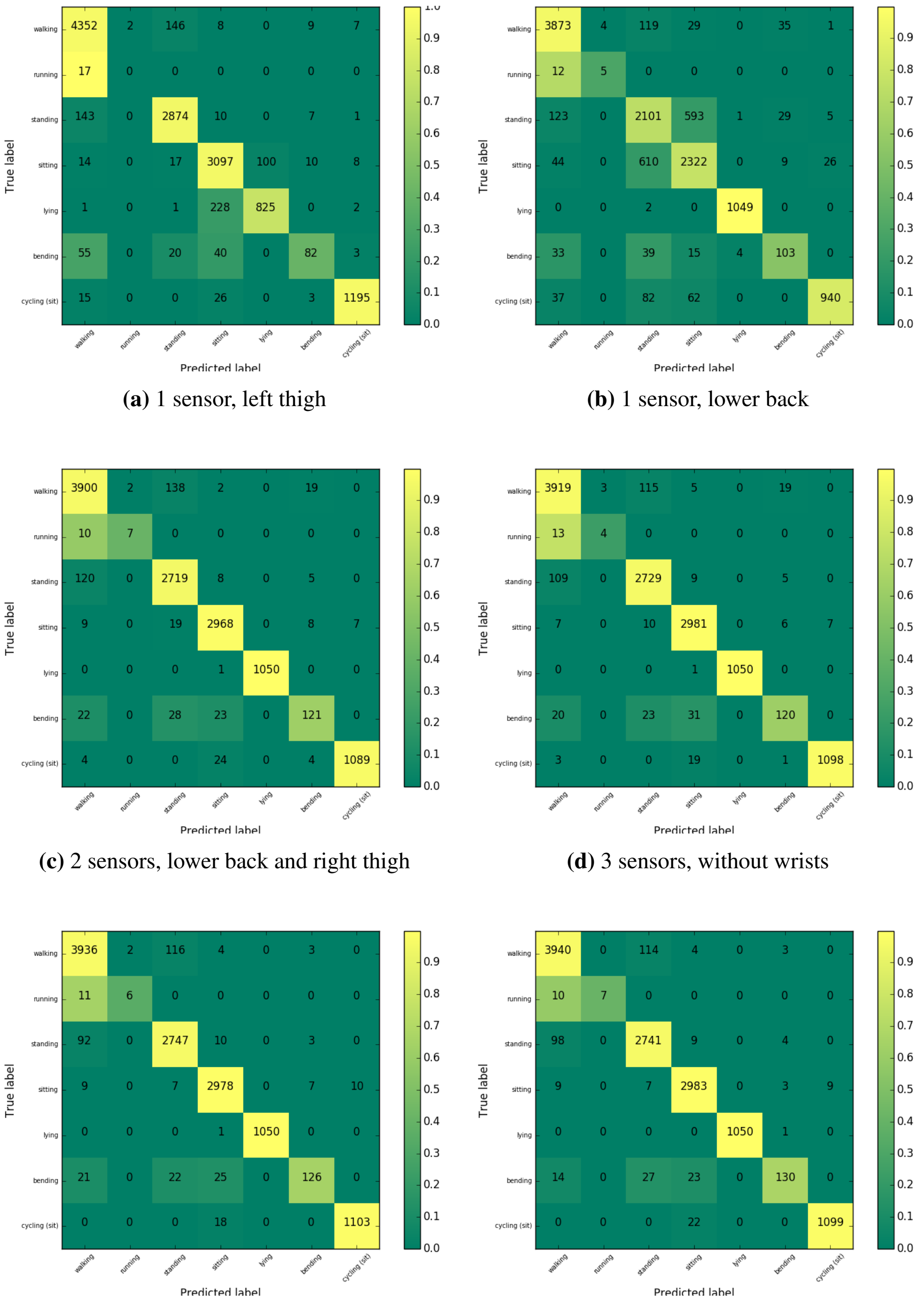
	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	3801	4	244	4	0	8	0
running	17	0	0	0	0	0	0
standing	91	0	2749	9	0	3	0
sitting	10	0	15	2973	0	7	6
lying	0	0	0	1	1050	0	0
bending	23	0	30	35	0	106	0
cycling (sit)	18	0	49	78	0	5	971

(e) 4 sensors, all except left wrist

	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	3800	2	247	5	0	7	0
running	17	0	0	0	0	0	0
standing	96	0	2743	8	0	5	0
sitting	12	0	14	2976	0	1	8
lying	0	0	0	0	1050	1	0
bending	20	0	23	33	1	117	0
cycling (sit)	75	0	1	119	0	0	926

(f) 5 sensors, all sensors

Figure 6.3: Confusion matrices for the best performing LOSO classifiers in section 6.2



(a) 1 sensor, left thigh

(b) 1 sensor, lower back

(c) 2 sensors, lower back and right thigh

(d) 3 sensors, without wrists

(e) 4 sensors, all except left wrist

(f) 5 sensors, all sensors

Figure 6.4: Confusion matrices for the best performing Mix-in classifiers in section 6.2

Table 6.3: Statistics for the best performing LOSO classifiers in section 6.2

(a) 1 sensor, left thigh

Activity	F_1	Rec.	Prec.	Spec.
walking	0.935	0.938	0.931	0.964
running	0.000	0.000	0.000	1.000
standing	0.924	0.940	0.909	0.972
sitting	0.825	0.927	0.743	0.897
lying	0.218	0.145	0.441	0.984
bending	0.490	0.380	0.688	0.997
cycling	0.930	0.882	0.983	0.998

(b) 1 sensor, lower back

Activity	F_1	Rec.	Prec.	Spec.
walking	0.922	0.920	0.924	0.963
running	0.026	0.018	0.050	1.000
standing	0.612	0.644	0.583	0.861
sitting	0.644	0.698	0.597	0.848
lying	0.997	0.998	0.995	1.000
bending	0.503	0.514	0.494	0.992
cycling	0.420	0.271	0.930	0.998

(c) 2 sensors, lower back and right thigh

Activity	F_1	Rec.	Prec.	Spec.
walking	0.939	0.926	0.952	0.977
running	0.009	0.006	0.017	1.000
standing	0.910	0.933	0.888	0.964
sitting	0.956	0.981	0.932	0.977
lying	0.999	0.999	0.999	1.000
bending	0.577	0.579	0.576	0.993
cycling	0.904	0.830	0.991	0.999

(d) 3 sensors, without wrists

Activity	F_1	Rec.	Prec.	Spec.
walking	0.943	0.929	0.957	0.980
running	0.000	0.000	0.000	1.000
standing	0.929	0.958	0.902	0.968
sitting	0.973	0.987	0.958	0.986
lying	0.999	0.999	1.000	1.000
bending	0.557	0.572	0.542	0.992
cycling	0.922	0.861	0.993	0.999

(e) 4 sensors, all except left wrist

Activity	F_1	Rec.	Prec.	Spec.
walking	0.935	0.927	0.943	0.972
running	0.059	0.035	0.200	1.000
standing	0.926	0.957	0.897	0.967
sitting	0.966	0.989	0.945	0.981
lying	0.999	0.999	0.999	1.000
bending	0.692	0.606	0.807	0.998
cycling	0.897	0.818	0.994	1.000

(f) 5 sensors, all sensors

Activity	F_1	Rec.	Prec.	Spec.
walking	0.938	0.930	0.947	0.974
running	0.000	0.000	0.000	1.000
standing	0.929	0.961	0.898	0.967
sitting	0.969	0.989	0.950	0.983
lying	0.999	0.999	0.999	1.000
bending	0.730	0.619	0.891	0.999
cycling	0.907	0.834	0.993	0.999

Table 6.4: Statistics for the best performing Mix-in classifiers in section 6.2

(a) 1 sensor, left thigh

Activity	$\mathbf{F}_1$	Rec.	Prec.	Spec.
walking	0.954	0.960	0.947	0.972
running	0.000	0.000	0.000	1.000
standing	0.942	0.946	0.938	0.982
sitting	0.929	0.949	0.909	0.970
lying	0.828	0.788	0.873	0.990
bending	0.526	0.411	0.731	0.998
cycling	0.975	0.964	0.985	0.999

(b) 1 sensor, lower back

Activity	$\mathbf{F}_1$	Rec.	Prec.	Spec.
walking	0.945	0.952	0.939	0.969
running	0.430	0.371	0.552	1.000
standing	0.727	0.744	0.711	0.909
sitting	0.769	0.767	0.771	0.926
lying	0.997	0.998	0.996	1.000
bending	0.550	0.536	0.565	0.993
cycling	0.894	0.830	0.970	0.997

(c) 2 sensors, lower back and right thigh

Activity	$\mathbf{F}_1$	Rec.	Prec.	Spec.
walking	0.959	0.959	0.960	0.981
running	0.349	0.282	0.471	1.000
standing	0.944	0.954	0.934	0.980
sitting	0.983	0.984	0.982	0.994
lying	0.999	0.999	0.999	1.000
bending	0.720	0.661	0.792	0.997
cycling	0.983	0.974	0.992	0.999

(d) 3 sensors, without wrists

Activity	$\mathbf{F}_1$	Rec.	Prec.	Spec.
walking	0.963	0.963	0.964	0.982
running	0.279	0.224	0.381	1.000
standing	0.952	0.959	0.945	0.983
sitting	0.984	0.989	0.980	0.994
lying	0.999	0.999	1.000	1.000
bending	0.718	0.649	0.804	0.997
cycling	0.987	0.979	0.995	1.000

(e) 4 sensors, all except left wrist

Activity	$\mathbf{F}_1$	Rec.	Prec.	Spec.
walking	0.966	0.967	0.965	0.983
running	0.738	0.594	1.000	1.000
standing	0.953	0.959	0.947	0.984
sitting	0.986	0.990	0.983	0.994
lying	0.999	0.999	1.000	1.000
bending	0.776	0.693	0.882	0.999
cycling	0.989	0.982	0.995	1.000

(f) 5 sensors, all sensors

Activity	$\mathbf{F}_1$	Rec.	Prec.	Spec.
walking	0.969	0.969	0.969	0.985
running	0.761	0.629	0.983	1.000
standing	0.955	0.963	0.948	0.984
sitting	0.987	0.990	0.983	0.994
lying	0.999	0.999	1.000	1.000
bending	0.801	0.700	0.936	0.999
cycling	0.989	0.983	0.994	0.999

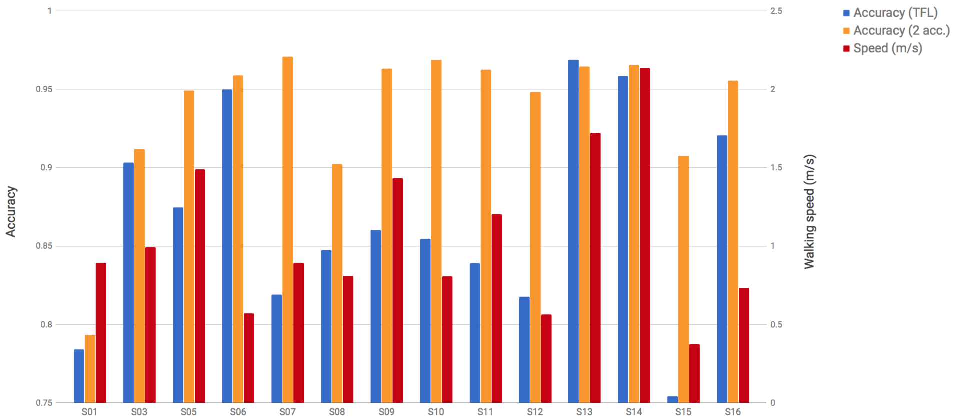


Figure 6.5: Accuracies achieved using the TFL classifier from section 6.1 (blue bars) and a LOSO classifier trained on stroke patient data (yellow bars). Walking speed is shown as red bars.

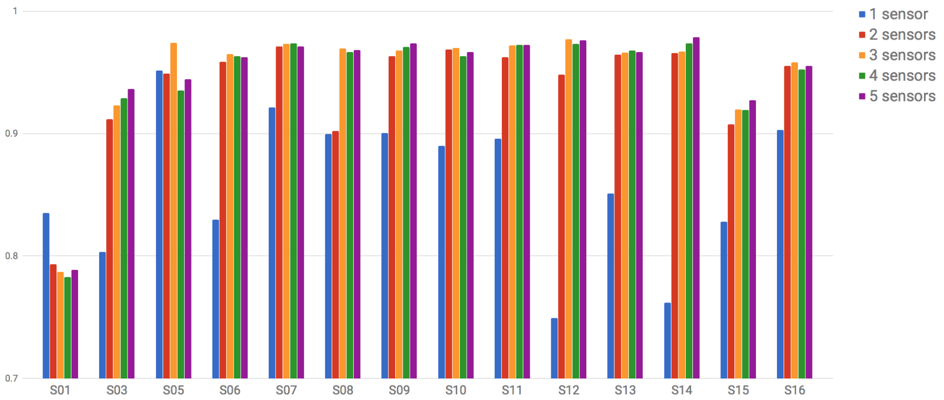


Figure 6.6: The increase in accuracy of the LOSO classifier for the best performing sensor configuration in section 6.2 as the number of sensors increases.

Cycling is the activity which is most affected by using training data from stroke patients instead of healthy subjects. This is seen by comparing figure 6.3c with figure 6.1d. Comparing tables 6.1b and 6.3c, the F_1-score for *cycling* has gone up from 0.393 to 0.904. Somewhat surprisingly, the F_1-score for *bending* has dropped from 0.587 to 0.577, mostly due to a lower recall. Adding *bending* samples from healthy subjects to the training set may be beneficial for this activity. The F_1-score for all other activities has gone up, with the exception of *running*. However, we will not dwell on these results, as the activity was only performed for a short duration by subjects S09 and S13 (see appendix C).

Physical ability has less of an impact on accuracy in this experiment than in the previous, as can be seen in figure 6.5, which shows the accuracies for each subject using the TFL classifier from section 6.1 (blue bars), the best two-sensor LOSO classifier in this experiment (yellow bars), and walking speed (red bars). Accuracy has increased dramatically for some of the less physically able subjects, but less so for more physically able subjects (e.g. compare subjects S03 and S12).

Subject S13 is the only subject for which the accuracy decreased slightly. S13 is the second most physically able subjects in the set. S13 was one of the two subjects who had samples labeled as *running* in its data set. The decrease may be due to the LOSO classifier misclassifying its *running* samples.

Personalized Models

The Mix-in models performed better than the semipopulation models for all sensor configurations, except when using only the right thigh sensor, where MP performed better than Mix-in (see row two of table 6.2a). This discussion will therefore focus on Mix-in classifiers. Semipopulation classifiers will be discussed in their own subsection.

If one has access to labeled data for a subject, a Mix-in classifier will be more accurate than a non-personalized classifier for any combination of sensors, as seen in table 6.2. The accuracy difference from the best to the worst performing Mix-in classifiers for each number of sensors is smaller than for non-personalized classifiers (compare for example the best and worst combinations in the LOSO and Mix-in columns in table 6.2b).

Changing sensor placements for subjects who cannot wear sensors on certain body parts seems to have less of an impact for Mix-in classifiers than non-personalized classifiers. We draw this conclusion because a higher share of the Mix-in classifiers for a given number of sensors achieve accuracies within 2–3 percentage points of the best classifier's accuracy than the non-personalized LOSO classifiers. One of the conclusions in the discussion of adapting the sensor placements for non-personalized classifiers was that the lower back sensor could not be removed without the accuracy going below 90%. Many Mix-in classifiers achieve more than 90% accuracy even without a lower back sensor. Making a Mix-in classifier for a user who cannot wear a lower back sensor is therefore an option if it is deemed to be worth the costs associated with labeling.

Automatically labeling the calibration set for a Mix-in classifier can be an option in a limited number of cases. Comparing figures 6.3a and 6.4a, it seems that automatically generated labels for a thigh sensor could help the classifier distinguish *lying* and *sitting*, as this is less of a problem for the Mix-in classifier than the LOSO classifier. However, because the automatically labeled data would have to come from a non-personalized classifier, we cannot expect much of a benefit if we use automatically labeled data to train classifiers that will classify data from more than one sensor: Comparing the confusion matrices in figures 6.3c through 6.3f and 6.4c through 6.4f, the Mix-in classifiers' advantage over the LOSO classifiers is less confusion of *walking* and *standing* and fewer misclassifications of *cycling* as *walking* and *sitting*. If automatically generated labels are to be used, many of the samples which would be mixed in with the training set would contain the misclassifications seen in the LOSO confusion matrices. This would not give much of a benefit to the Mix-in classifiers during training.

Bending is nearly as problematic for the Mix-in classifiers as it was for the LOSO classifiers. Looking at the confusion matrices, the Mix-in classifiers are only capable of correctly classifying about ten additional samples of *bending* throughout all subjects. This indicates an underlying problem with the activity other than each subject performing the activity differently, possibly a very high variation in how the activity is performed even by a single subject, insufficient features to distinguish it from the other activities, or, probably, an unclear or overlapping definition. The definition of the activity, seen in appendix B, includes both bending when sitting and standing, which may explain why the activity is often confused with *sitting* and *standing*. The many misclassifications of *bending* as *walking* may be due to the subject re-positioning his or her feet during the activity. It may be beneficial to classification to separate *bending* into two classes: *bending (sitting)* and *bending (standing)*.

Semipopulation Classifiers as an Alternative to Ordinary Personalized Models

Comparing the Mix-in, MP, and SP columns in table 6.2, the results are not in favor of the semipopulation approaches: Neither MP nor SP classifiers achieve better results than mix-in classifiers (except in row two of table 6.2a, where MP outperforms Mix-in). This makes it hard to recommend semipopulation classifiers as a technique for making personalized classifiers.

Nonetheless, it is impressive that the semipopulation classifiers perform as well as they do, given the limited set of hypotheses (i.e. sub-models) that the technique gets to choose from during the calibration phase. For example, the SP classifiers lead to higher accuracies than the LOSO classifiers in nearly all rows of tables 6.2a and 6.2b. The SP classifiers' sub-models are all trained on data from just one individual, while the LOSO classifier is trained on data from this individual *and* all other individuals in the training set (except the test subject). If some way of knowing which individual would be picked by the SP calibration without the need for a calibration set was discovered, using SP models could turn out to be a superior alternative to a general, non-personalized model.

A later experiment will look at whether these techniques can be used as a tool for diagnosis, an area in which these techniques may still be useful.

6.3 Sensor Matching by Affected Side

Section 2.6.1 explained how stroke patients often experience weakness or paralysis in one side of the body. It is therefore likely that we can find more similarities between subjects if we match their sensors by body part and their body's affected and unaffected side, rather than left and right side.

This experiment's hypothesis is that matching by affected sides will have a positive impact on performance. We will be most interested in the effects on sensor combinations which have already been shown to lead to good performance (more than 90% accuracy) with non-personalized models. These combinations all contained one lower back sensor and one or two thigh sensors. We expect sensors on the unaffected side to be more beneficial to classification than their unaffected counterparts as these are more involved in movements.

For the sake of completeness, we will still test the worse-performing combinations and semipopulation classifiers, in the case that the matching scheme should lead to improved performance for them.

6.3.1 Setup

Affected side has been registered for all patients in the TCS data set. Ten of the fifteen subjects have an affected left side. The experiment in section 6.2 will be repeated, matching the sensors by affected and unaffected sides rather than left and right sides. All the same sensor combinations and classification approaches will be used. The impact of this on the classification task will be measured. The experiment will be run ten times, using the same seed values as in the previous experiment in order to get identical test sets.

Matching sensors by affected side may lead to sensors with opposite axis directions in relation to movement and gravity direction being compared. The steps taken to ensure that this does not affect the final values of the features were explained in section 5.3.2.

6.3.2 Results and Discussion

Table 6.5 shows the accuracy scores achieved using the different combinations[1]. Figure 6.9 shows confusion matrices when using only the affected and unaffected thigh. A plot comparing subject accuracy for the best performing two-sensor classifier from the previous and this experiment is seen in figure 6.7. Figure 6.8 shows the best-classifier accuracy as the number of sensors increases. Figure 6.10 shows the confusion matrices for the best two-sensor classifier in the previous and this experiment, and table 6.6 their respective statistics.

Contrary to expectations, matching sensors by affected side does not lead to significant improvements for most sensor combinations that already result in accuracies above 90% using non-personalized classifiers. With the exception of table 6.5b, none of tables' best-performing classifiers show a significantly improved accuracy compared to the best classifiers in table 6.2.

The matching scheme has no clear effect on Mix-in classifiers. Improvements in the semipopulation classifiers are not so large that they can be considered an alternative to Mix-in classifiers. We will therefore not discuss these classifiers in any more detail.

Opposite of what we assumed, the affected thigh seems to be a better sensor than the unaffected thigh in almost any sensor combination in table 6.5. Figure 6.9 shows confusion matrices when using each of these sensors exclusively. The affected thigh leads to fewer

[1] The third row of table 6.5a is identical to the third row of table 6.2a because the lower back sensor does not match with a different sensor when affected-side-matching is used instead of left-right-matching, leading to identical training and test sets.

Table 6.5: Accuracy scores from section 6.3's experiment. Percentage point difference to the combination at the same position in table 6.2 shown in parentheses, rounded to one significant digit.

(a) 1 sensor.

lb	at	ut	aw	uw	LOSO	Mix-in	MP	SP
	x				85.64 (-0.2)	93.00 (-0.2)	92.87 (0.2)	89.70 (1.7)
		x			85.39 (0.4)	92.13 (0.3)	93.17 (0.3)	88.92 (0.2)
x					74.17 (—)	84.40 (—)	78.85 (—)	75.81 (—)
				x	60.99 (0.9)	70.90 (-1.7)	63.96 (-2.4)	60.46 (-1.9)
			x		55.73 (0.6)	73.41 (2.7)	67.16 (2.5)	60.81 (2.5)

(b) 2 sensors

lb	at	ut	aw	uw	LOSO	Mix-in	MP	SP
x	x				93.60 (0.4)	96.30 (—)	94.75 (-0.4)	93.27 (-0.4)
x		x			93.11 (—)	96.23 (—)	94.58 (0.1)	93.38 (0.5)
	x	x			86.88 (-1.0)	94.42 (0.4)	93.54 (0.9)	90.60 (2.5)
	x			x	86.55 (-0.1)	94.02 (-0.8)	92.44 (-0.9)	89.37 (-1.4)
		x		x	85.86 (0.8)	93.23 (-1.2)	92.89 (0.7)	88.91 (2.1)
	x		x		85.16 (0.1)	94.35 (1.1)	92.30 (0.3)	87.67 (-0.7)
		x	x		85.06 (0.1)	93.43 (0.1)	92.00 (0.4)	87.23 (-0.3)
x				x	78.74 (1.3)	87.93 (-0.6)	80.70 (0.2)	77.56 (0.7)
x			x		74.18 (-0.8)	87.77 (0.6)	78.17 (0.8)	74.33 (-0.3)
			x	x	64.32 (-1.1)	79.44 (-0.3)	68.44 (-0.3)	61.99 (1.0)

(c) 3 sensors

lb	at	ut	aw	uw	LOSO	Mix-in	MP	SP
x	x	x			94.15 (-0.1)	96.71 (—)	95.29 (0.2)	93.92 (—)
x	x			x	93.84 (0.3)	96.78 (0.1)	94.67 (0.1)	93.66 (0.7)
x		x		x	93.48 (—)	96.67 (-0.1)	94.14 (-0.2)	92.72 (-0.6)
x	x		x		93.18 (0.6)	96.63 (—)	94.37 (0.6)	93.24 (0.9)
x		x	x		92.60 (0.1)	96.55 (—)	93.73 (-0.6)	91.70 (-1.1)
	x	x	x		88.00 (0.6)	95.29 (0.2)	93.28 (-0.4)	89.13 (-1.3)
	x	x		x	87.30 (0.1)	95.14 (-0.4)	94.14 (0.9)	90.00 (—)
	x		x	x	85.76 (-0.9)	94.84 (0.3)	92.08 (0.3)	87.85 (1.7)
		x	x	x	85.36 (-0.1)	93.88 (0.1)	91.60 (0.4)	87.07 (-0.4)
x			x	x	78.51 (-1.2)	89.16 (-0.3)	79.43 (0.5)	75.81 (1.0)

(d) 4 sensors

lb	at	ut	aw	uw	LOSO	Mix-in	MP	SP
x	x	x		x	94.16 (-0.4)	97.07 (—)	95.32 (0.1)	94.13 (0.6)
x	x	x	x		94.05 (0.1)	97.01 (—)	94.91 (—)	94.12 (0.5)
x	x		x	x	93.36 (0.3)	96.94 (0.1)	94.23 (0.4)	93.50 (1.4)
x		x	x	x	92.89 (-0.1)	96.76 (-0.1)	93.80 (0.3)	91.45 (-1.0)
	x	x	x	x	87.86 (0.5)	95.54 (—)	93.39 (0.1)	89.01 (-0.5)

(e) 5 sensors

lb	at	ut	aw	uw	LOSO	Mix-in	MP	SP
x	x	x	x	x	94.09 (-0.2)	97.18 (—)	95.06 (—)	93.97 (0.5)

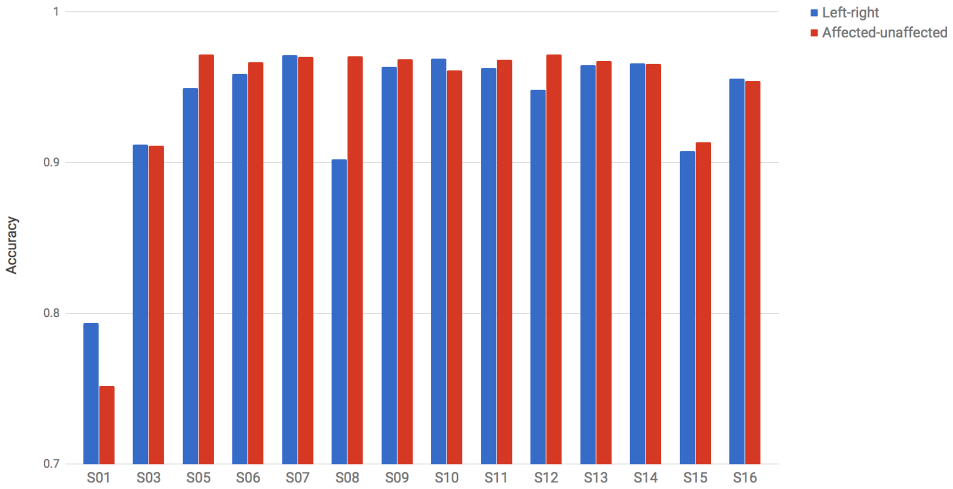


Figure 6.7: Difference between the best performing two-sensor LOSO classifier in section 6.2 and the best performing two-sensor LOSO classifier in section 6.3.

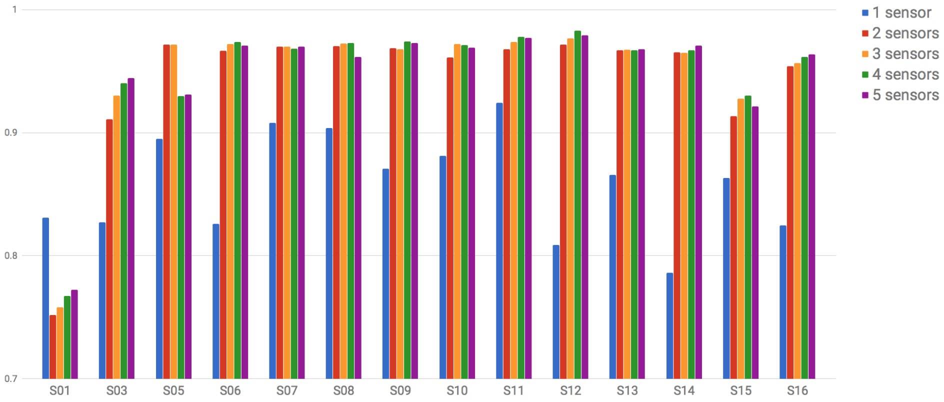
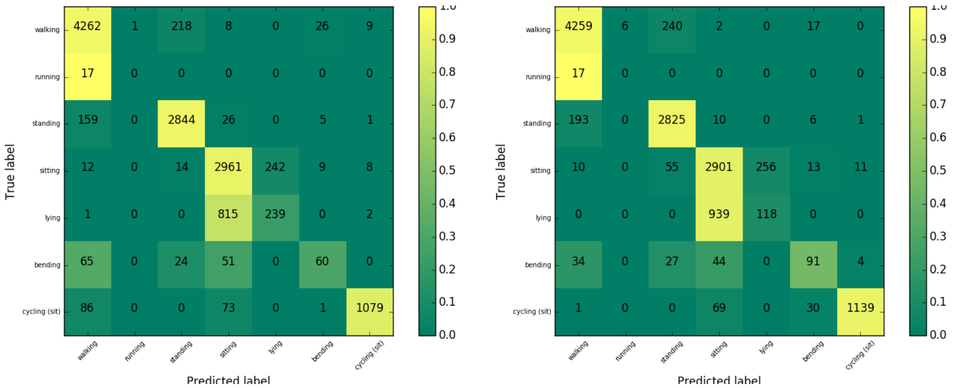


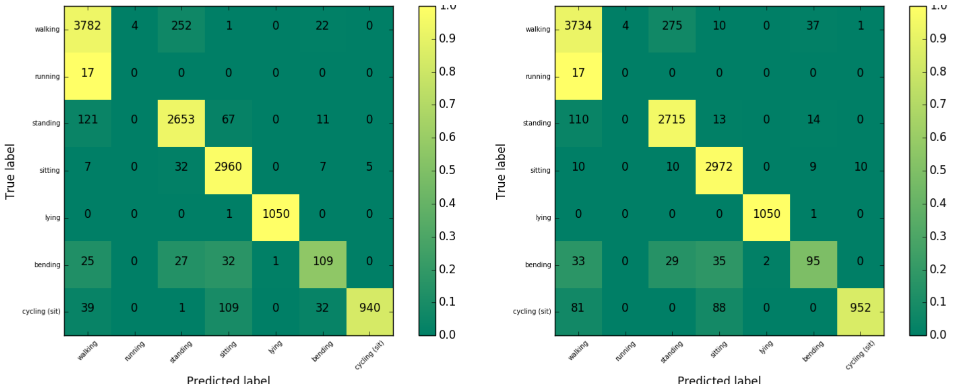
Figure 6.8: The increase in accuracy of the best LOSO classifier for each number of sensors in section 6.3 as number of sensors increases.



(a) 1 sensor, affected thigh

(b) 1 sensor, unaffected thigh

Figure 6.9: Confusion matrices from one run using the affected and unaffected thigh sensors with a LOSO classifier



(a) 2 sensors, lower back and right thigh (repeated from page 84)

(b) 2 sensors, lower back and affected thigh

Figure 6.10: Confusion matrices from one run for the best performing two-sensor LOSO classifiers from the previous and this experiment

Table 6.6: Confusion matrices from one run for the best performing two-sensor LOSO classifiers from the previous and this experiment

(a) 2 sensors, lower back and right thigh (repeated from page 86)

Activity	F_1	Rec.	Prec.	Spec.
walking	0.939	0.926	0.952	0.977
running	0.009	0.006	0.017	1.000
standing	0.910	0.933	0.888	0.964
sitting	0.956	0.981	0.932	0.977
lying	0.999	0.999	0.999	1.000
bending	0.577	0.579	0.576	0.993
cycling	0.904	0.830	0.991	0.999

(b) 2 sensors, lower back and affected thigh

Activity	F_1	Rec.	Prec.	Spec.
walking	0.928	0.918	0.938	0.970
running	0.000	0.000	0.000	1.000
standing	0.923	0.953	0.896	0.966
sitting	0.971	0.987	0.956	0.985
lying	0.999	0.999	0.998	1.000
bending	0.549	0.512	0.592	0.994
cycling	0.917	0.852	0.991	0.999

misclassifications of *sitting* and *lying*. Many samples of these activities are misclassified as *standing*. It could be that the affected part of the body is more at rest when performing these activities, making them easier to distinguish. The affected thigh is more beneficial for *bending*, but as stated in the discussion of the previous experiment, the lower back sensor complements the thigh sensor in this activity.

Accuracy using the best two-sensor combination has improved. A plot comparing subject accuracy for the best performing two-sensor classifiers from the previous and this experiment is seen in figure 6.7. Subjects S05, S08, and S12 have improved significantly. Subject S01 is negatively affected, but this subject is already known to be problematic. Figure 6.8 shows the best-classifier accuracy for each number of sensors. Accuracy increases gradually for nearly all subjects as the number of sensors increase, the main exception being S05. Figure 6.10 shows the confusion matrices for the best two-sensor classifier in the previous and this experiment. Affected side matching is beneficial for classifying *standing* and *sitting*, but slightly detrimental to recognizing *walking*. These observations are confirmed by comparing the activities' F_1-scores in table 6.6.

From the results in this experiment, affected side matching can be seen as a reasonable alternative to matching by left and right sides if two sensors are to be used, as it was beneficial to accuracy, especially for subject S08. The matching scheme should depend on what activities are seen as most important to recognize. As a patient can be expected to spend more time sitting than standing or walking, we give a slight recommendation to using affected side matching.

6.4 Mixing Stroke Patient and Healthy Subject Training Data

Section 6.1 investigated the performance of a classifier trained exclusively on training data from healthy subjects (the TFL data set) when tested on stroke patients (the TCS data set) using one thigh and one back sensor. This classifier achieved 86.5% accuracy on the TCS data set, compared to 94.2% on its own data set. Experiments in sections 6.2 and 6.3 showed that the system could achieve 93% accuracy and higher on the same test set when trained on stroke patient data using the same set of sensors.

In this experiment, we will train classifiers on a training set consisting of training data from both of these groups and test them on their subjects in a LOSO fashion. We will look at what effects this has on overall accuracy and on the individual activities for both the healthy subjects and stroke subjects. It would be interesting to find that this is beneficial or only slightly detrimental to the classification quality for stroke patients: Stroke patients have different levels of physical ability and will hopefully improve in their ability over time. Not having to change the classifier as a subject makes progress will make it easier to follow a subject's progress over time. For healthy subjects, a positive effect will also be interesting. It could be that data from a more varied pool of individuals is beneficial when the classifier has to handle variations in activity performance.

Results could be positive for some activities and negative for the rest. This could indicate that it would be beneficial to use training data from healthy subjects for some activities when making a classifier meant exclusively for stroke patients and vice versa.

Section 6.3 showed that affected side matching was beneficial for classifying stroke patient data using one thigh and one back sensor. We will therefore use the affected matching scheme for stroke patients. To see whether the unaffected or affected side is best for this purpose, we will perform two versions of the experiment: One using the affected thigh sensor and one using the unaffected thigh sensor. Healthy subjects wore only one thigh sensor, and this will be used in both versions.

6.4.1 Setup

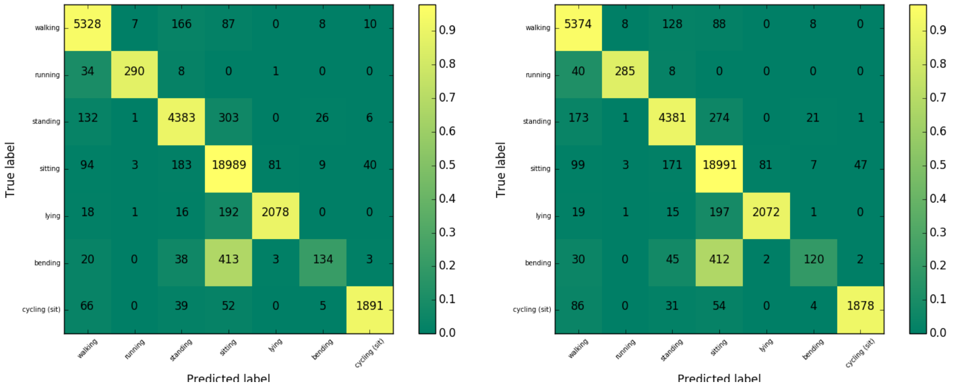
A classifier will be trained on a training set consisting of the TFL and TCS data sets and tested on all individuals in each training set in a LOSO fashion. One thigh and one back sensor from each subject will be used, and two separate versions of the experiment will be run, one using the unaffected thigh sensor and one using the affected thigh sensor. Both of these versions will be run ten times, and the results presented will be averages over these runs.

6.4.2 Results and Discussion

Table 6.7 shows the accuracies achieved on the two training sets using the two matching schemes. Figures 6.11 and 6.12 shows confusion matrices displaying how activities in the two test sets, using the unaffected thigh sensor, were classified before and after training data from the other set was mixed in. Tables 6.8 and 6.9 show statistics for the same training and test sets.

Matching scheme / Test set	Affected	Unaffected
Trondheim Free Living	94.37	94.35
Trondheim Chronic Stroke	92.10	93.22

Table 6.7: Accuracy achieved when using both stroke patient and healthy subjects training data in the training set.



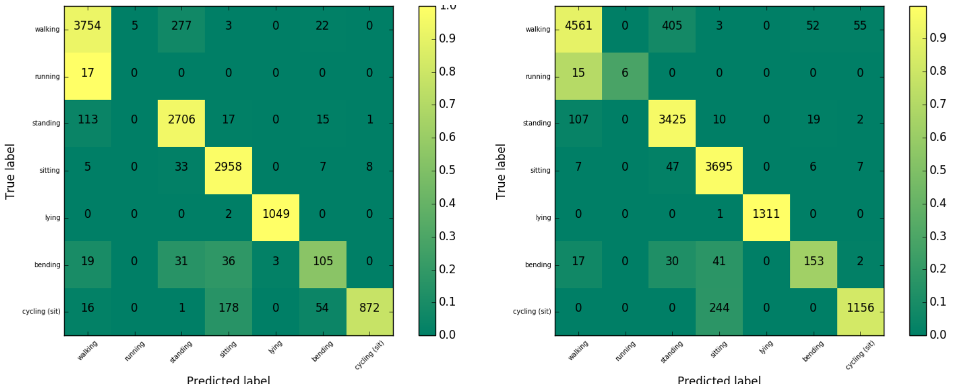
	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	5328	7	166	87	0	8	10
running	34	290	8	0	1	0	0
standing	132	1	4383	303	0	26	6
sitting	94	3	183	18989	81	9	40
lying	18	1	16	192	2078	0	0
bending	20	0	38	413	3	134	3
cycling (sit)	66	0	39	52	0	5	1891

(a) Classifying TFL without stroke patient data (repeated from page 77)

	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	5374	8	128	88	0	8	0
running	40	285	8	0	0	0	0
standing	173	1	4381	274	0	21	1
sitting	99	3	171	18991	81	7	47
lying	19	1	15	197	2072	1	0
bending	30	0	45	412	2	120	2
cycling (sit)	86	0	31	54	0	4	1878

(b) Classifying TFL with stroke patient data

Figure 6.11: Confusion matrices from one run when classifying the TFL subjects with or without stroke data, unaffected thigh sensor used.



	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	3754	5	277	3	0	22	0
running	17	0	0	0	0	0	0
standing	113	0	2706	17	0	15	1
sitting	5	0	33	2958	0	7	8
lying	0	0	0	2	1049	0	0
bending	19	0	31	36	3	105	0
cycling (sit)	16	0	1	178	0	54	872

(a) Classifying TCS without healthy patient data, from experiment 6.3

	walking	running	standing	sitting	lying	bending	cycling (sit)
walking	4561	0	405	3	0	52	55
running	15	6	0	0	0	0	0
standing	107	0	3425	10	0	19	2
sitting	7	0	47	3695	0	6	7
lying	0	0	0	1	1311	0	0
bending	17	0	30	41	0	153	2
cycling (sit)	0	0	0	244	0	0	1156

(b) Classifying TCS with healthy patient data

Figure 6.12: Confusion matrices from one run when classifying the TCS subjects with or without healthy data, unaffected thigh sensor used. Note that total number of samples for each activity is different because 20% of the test data for each subject was removed from LOSO testing in experiments 6.2 and 6.3 (see section 6.2.1 for explanation).

Table 6.8: Statistics when classifying the TFL subjects with or without stroke data, unaffected thigh sensor used.

(a) Classifying TFL subjects using LOSO (repeated from page 78)

Activity	F_1	Rec.	Prec.	Spec.
walking	0.944	0.951	0.938	0.988
running	0.913	0.871	0.958	1.000
standing	0.905	0.902	0.908	0.985
sitting	0.963	0.979	0.948	0.934
lying	0.938	0.915	0.962	0.997
bending	0.334	0.216	0.731	0.999
cycling	0.948	0.925	0.973	0.998

(b) With stroke patient training data

Activity	F_1	Rec.	Prec.	Spec.
walking	0.942	0.960	0.925	0.985
running	0.908	0.864	0.956	1.000
standing	0.910	0.903	0.917	0.987
sitting	0.965	0.979	0.952	0.939
lying	0.943	0.925	0.962	0.997
bending	0.311	0.196	0.752	0.999
cycling	0.943	0.915	0.973	0.998

Table 6.9: Statistics when classifying the TCS subjects with or without stroke data, unaffected thigh sensor used.

(a) Without healthy training data

Activity	F_1	Rec.	Prec.	Spec.
walking	0.939	0.925	0.954	0.978
running	0.051	0.035	0.091	0.999
standing	0.919	0.951	0.890	0.965
sitting	0.954	0.982	0.928	0.975
lying	0.999	0.999	0.998	1.000
bending	0.572	0.576	0.570	0.993
cycling	0.871	0.777	0.991	0.999

(b) With healthy training data

Activity	F_1	Rec.	Prec.	Spec.
walking	0.936	0.904	0.971	0.987
running	0.508	0.348	1.000	1.000
standing	0.918	0.963	0.877	0.959
sitting	0.952	0.981	0.925	0.974
lying	0.999	0.999	1.000	1.000
bending	0.658	0.624	0.697	0.996
cycling	0.882	0.823	0.950	0.996

The unaffected thigh sensor is the best of the two thigh sensors if we are to make a general classifier for both groups, as seen in table 6.7. In section 6.3, the affected thigh was shown to lead to a higher accuracy, but that experiment's data set consisted only of stroke patients. Because the unaffected side moves more than the affected side, it is not surprising that healthy subjects and stroke patients look more similar to the classifier when the unaffected thigh is used. Classifiers trained with this matching scheme leads to an improvement of about 0.1 percentage points for both test sets, judging from the results in table 6.7 compared to previous results on the same test sets.

The confusion matrices for classifiers trained with the unaffected thigh sensor, seen in figures 6.11 and 6.12, show that there is no large overall difference for neither stroke patients nor healthy subjects when using training data also from the other group. Training a general classifier which can classify individuals from both of these groups can therefore be considered a safe choice. The notion that it is safe to make a general classifier for both these groups is confirmed by the activities' F_1-scores in tables 6.8 and 6.9, which are only changed marginally when using a training data set from both groups. One exception to this is *running*, which has a much higher score for stroke patients when healthy training data is mixed into the training set. This is not surprising, as the running samples in the TCS data set are so few that the non-personalized classifiers trained exclusively on stroke patient data were rarely able to recognize the activity in the previous experiments.

Future experiments with classifiers aimed specifically at stroke patients or healthy subjects should evaluate whether it is beneficial to use training data from the other group for specific activities. This is because some activities' F_1-scores are affected negatively by mixing the groups' samples while others are affected positively. For example, classifiers for stroke patients could benefit from having *bending* and *cycling* samples from healthy subjects (deleting the healthy subject samples for other activities). Classifiers for healthy subjects could benefit from having *standing* samples from stroke patients.

6.5 Semipopulation Models as a Tool for Finding Subject Similarities

The calibration phase in the semipopulation approaches (explained in section 5.7.3) is based on connecting a new user to existing users' sub-models. The subjects and the selected sub-models can be viewed as a graph: Subjects could be seen as nodes. The calibration procedure choosing one of some subject B's sub-models to classify one of some subject A's activities could be seen as a directed edge from node A to node B.

Let us call the case where two subjects are connected by edges pointing back and forth a "reciprocal connection". Reciprocal connections occur when subject A's and subject B's sub-models are selected for the same activity in MP, or when the subjects' overall set of sub-models are mutually selected in SP. If the calibration phase is indeed guided by some underlying similarity between the users, we would expect that there is a considerably higher probability of reciprocal connections than would be predicted by chance.

Finding similarities between sub-models can be useful if it is shown to be related to some attribute of the users which would otherwise be costly to observe. For our purposes, it would be useful to be able to discern the wearer's physical ability, as measured by walking speed on TUG and 10 meter walk tests, from the connections, as physical ability would otherwise have to be measured by medical personnel. Finding that the similarity is dependent on attributes which are easy to measure with other tools, such as the user's weight and height, would be less useful.

Hypothetically, if this experiment finds that there is a connection between sub-model selection and physical ability, a system could first use a non-personalized classifier to get predictions for a week-long recording of a subject. The system would then use semipopulation calibration find out which sub-models would be selected for the subject based on the non-personalized classifier's predictions for this week. Afterwards, the system would give a report about the subject's assumed physical condition based on the physical ability of the subjects whose sub-models were chosen.

6.5.1 Setup

During the experiments in sections 6.2 and 6.3, the system logged which subjects' sub-models were selected in the calibration phases for all runs of the MP and SP classifiers. Exactly 620 SP and an equal number of MP runs were performed using all available subjects. These files will be analyzed for reciprocal connections, on an activity-by-activity basis for MP and on a subject-by-subject basis in SP. The number of actual reciprocal connections will be compared to the number of connections that would be expected by chance if the probabilities were uniformly distributed among the other individuals.

Each sensor's average number of reciprocal connections will also be examined, calculated as the total number of reciprocal connections in the runs that the sensor was involved in divided by the number of runs. We would expect that sensors which are already accurate for classification are also accurate in finding similarities in physical abilities.

Should the numbers indicate that the process is guided by some underlying similarity, the results from runs which yielded many reciprocal connections will be examined to see whether they seem to match up with the subjects' TUG and 10 meter walk scores.

6.5.2 Results and Discussion

Most of the semipopulation runs involved 14 subjects, as S02 was left out of any configuration involving a lower back sensor. Let us denote the probability that a network with 14 nodes has n reciprocal connections by $P_r(n)$, given that every node is equally likely to connect to any node in the network other than itself. A simulation of a 14 node network was run 10^6 times find the probability distribution P_r, resulting in $P_r(0) = 0.558$, $P_r(1) = 0.355$, $P_r(2) = 0.079$, $P_r(3) = 0.008$.

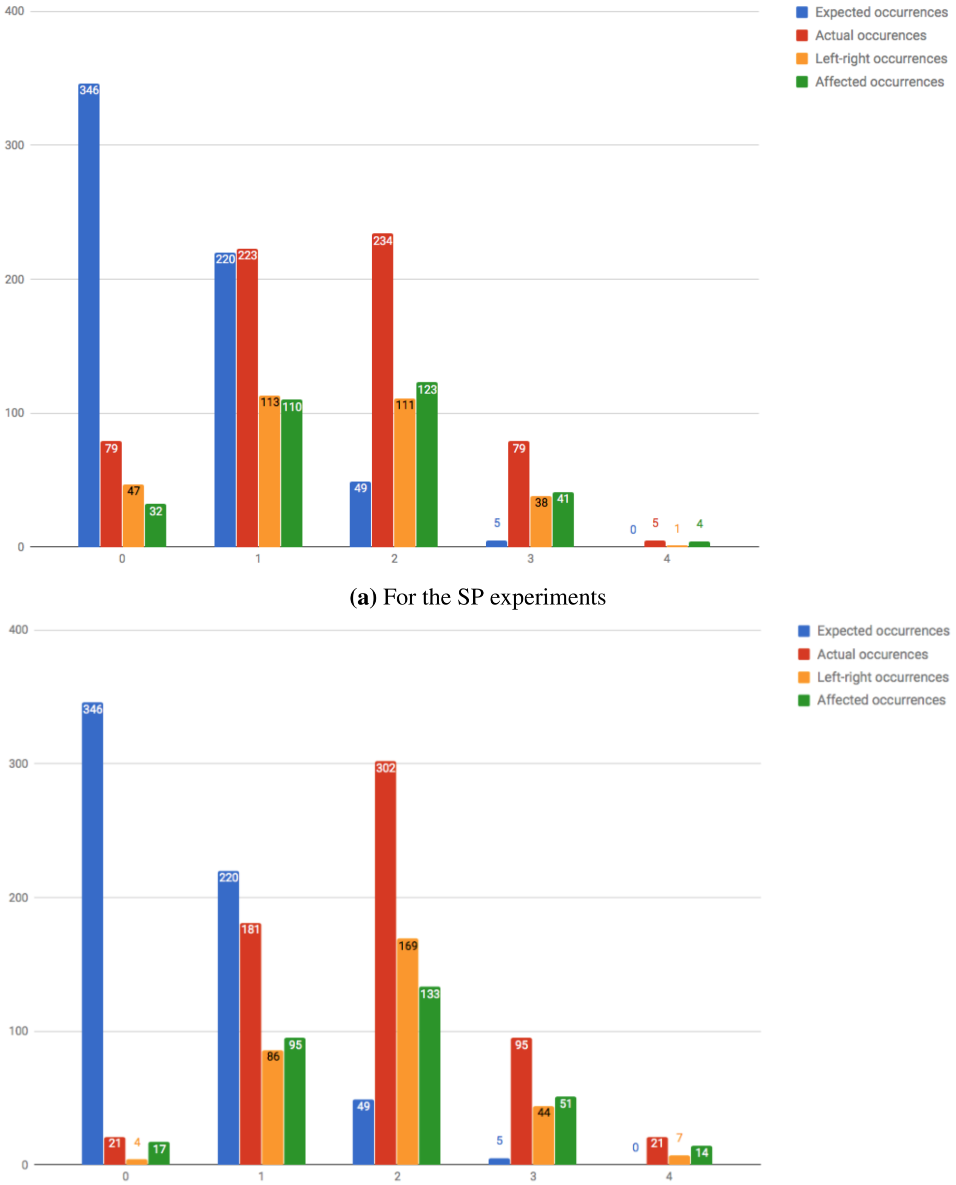
Figure 6.13a shows the expected and actual number of reciprocal connections in the SP calibrations, along with two additional columns showing how the reciprocal connections were divided between the left-right and affected matching schemes. An equivalent graph for the number of reciprocal connections using the *cycling* activity in the MP experiments is shown in figure 6.13b. Cycling is displayed because it was the only activity which achieved a higher average number of reciprocal connections than the SP experiments, but all activities performed better than chance. Looking at these graphs, the calibration procedure clearly does something other than picking individuals with a uniform distribution.

Figure 6.14 shows the average number of reciprocal connections by sensor placement. Looking at the numbers from the SP experiments, there seems to be some correlation between a sensor's utility in classification and its ability to lead to a high number of reciprocal connections: The thigh sensors lead to better accuracy and more reciprocal connections than the wrist sensors. Within the thigh sensor pairs, the most accurate sensors leads to the highest number of reciprocal connections, but the connection is the opposite for the wrist sensors. The lower back (lb) sensor is the second worst at generating reciprocal connections, which is a bit strange, as it is involved in the best sensor combinations whenever two or more sensors are used in tables 6.2 and 6.5. Possibly, the lower back sensor's lack of success in generating reciprocal connections is due to its strength in general: If the features extracted from it generalize well among the individuals, they may not serve to distinguish them. Examining the number of connections for the *cycling* activity in the MP classifiers, there is no clear connection between a sensor's utility in classification and leading to a higher number of reciprocal connections. In fact, the case seems to be almost the opposite: The affected wrist (aw), which was the single worst performing sensor in table 6.5a, is the best at leading to reciprocal connections. All in all, there is no apparent connection between accuracy in classification and the average number of reciprocal connections.

Graph plots of the runs which resulted in four reciprocal connections were examined. Subject nodes were shown as dots; blue dots are male and red dots female subjects. Edges, drawn as arrows, show the sub-models selected for the subjects. An arrow going out from node A with its arrowhead at node B means B's sub-model was selected for A in calibration. To see whether physical condition or physical attributes matter more to the connections, the graphs were plotted in two Cartesian coordinate systems: One with the subjects placed by their TUG and 10 meter walk scores, and the other with the subjects placed by their height and weight. The length of the arrows thus indicate how similar the individuals are in the attributes on the plot's axes.

Figures 6.15 and 6.16 show plots of the sensor configurations found to be most accurate in sections 6.2 and 6.3 which led to four reciprocal connections. The first thing to observe is that neither of the figures lend much credence to the idea that the connections are based on physical ability. While there is a cluster of highly connected subjects in the bottom left of figure 6.15a, the same subjects are represented in the top right of figure 6.15b. If anything, the comparison of figures 6.15a and 6.15b seems to speak for the connections being based on physical proportions. Comparing figures 6.16a and 6.16b seems to affirm that weight and height play a greater role than physical ability. A cause for skepticism is the lack of similarity between figures 6.15a and 6.16a, which should have many edges in common if physical ability was a clear cause of the connections. Similar plots of sensor configurations which were found to be more accurate in previous experiments, but yielded fewer reciprocal connections have been inspected. None of these plots would lead to any conclusion supporting the use of semipopulation calibration as a tool to estimate physical ability. There is also a lack of similarity between figures 6.15b and 6.16b, and there is therefore little reason to think that we can estimate height and weight using semipopulation approaches. Height and weight are so easily measured through other means that using semipopulation approaches to estimate this would be unnecessary.

Summarizing, the results in figure 6.13 point towards there being some underlying connection between individuals. As seen in figures 6.15 and 6.16, none of these connections seem to indicate that semipopulation strategies can be used to draw conclusions about the subjects' physical abilities. The results may indicate that the sub-models found in calibration are somewhat related to physical proportions such as weight and height. Future research should therefore investigate whether there is some benefit to classifying individuals with non-personalized classifiers trained on data from individuals with similar physical proportions. As the results of this experiment and the experiments in sections 6.2 and 6.3 have failed to show any benefit to using semipopulation approaches, there is no reason yet to recommend such classifiers for use outside of research.



(a) For the SP experiments

(b) For the *cycling* activity in the MP experiments

Figure 6.13: Distribution of the number of reciprocal connections in selected semipopulation experiments

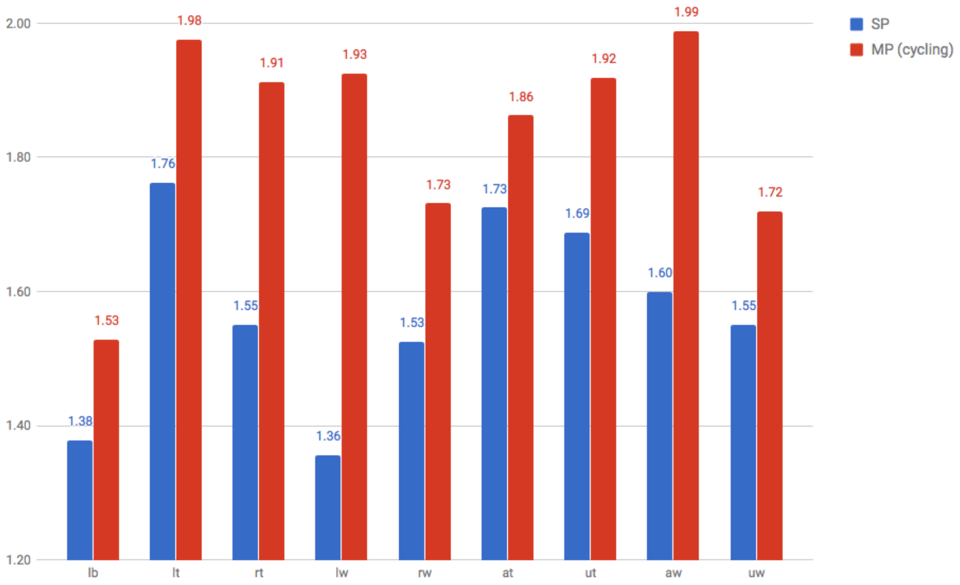
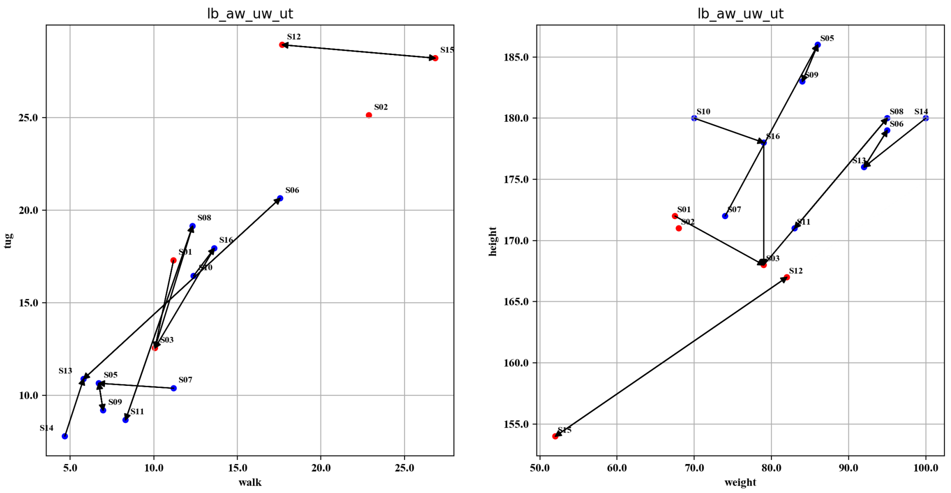


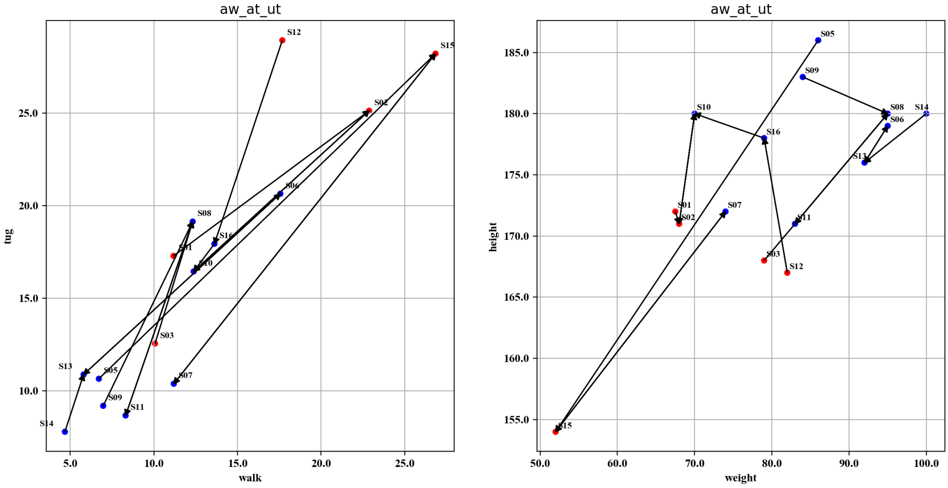
Figure 6.14: Average number of reciprocal connections by sensor



(a) By TUG and 10 meter walk scores

(b) By weight and height

Figure 6.15: Graph plots of the most accurate SP sensor configuration to yield 4 reciprocal connections between subjects overall.



(a) By TUG and 10 meter walk scores

(b) By weight and height

Figure 6.16: Graph plots of the most accurate MP sensor configuration to yield 4 reciprocal connections for the *cycling* activity.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis has resulted in a human activity recognition system which can be used to recognize *walking*, *standing*, *sitting*, *lying*, *bending* and *cycling* for stroke patients from accelerometer recordings. Using acceleration values from accelerometers on the subject's lower back and affected thigh, the system's non-personalized models were able to achieve 93.6% accuracy on average in a leave-one-subject-out evaluation. With additional accelerometers attached to the subject's other thigh and right wrist, the system achieved 94.6% accuracy on average in a similar evaluation. This performance has been shown to be on the same level as the system's performance when trained and tested on healthy subjects.

Subjects in the Trondheim Chronic Stroke data set wore sensors on five different body parts, each contributing approximately one hour of sensor data in a laboratory environment. Experiments in this thesis have used every possible combination of these sensors to train and test the system, including combinations that matched sensors by the subject's affected sides rather than left or right side. The outcome of these experiments showed that, if a patient cannot wear a sensor on a specific body part, there are many alternative sensor configurations which can be used instead, leading to accuracy scores within a few percentage points of the best results. Using both of the thigh sensors and the lower back sensor was found to be best for classification. Using one less thigh sensor, the system could still achieve about 93% accuracy. Adding one or two wrist sensors to this combination had a slight positive effect on accuracy.

The same experiments evaluated personalized models with access some of the test subject's labeled data in their training set. The purpose of these models is to be an additional alternative for patients who cannot wear sensors on specific body parts. The experiments in this thesis showed that personalized models using only a thigh sensor could achieve 93% accuracy, about the same as non-personalized models using a lower back and one thigh sensor. Non-personalized models could not achieve more than 90% accuracy using only one sensor, and, when two or more sensors were used, also needed to have a lower back sensor in the configuration to achieve more than 90% accuracy. Personalized models were able to use multiple-sensor combinations *without* the lower back sensor and still achieve more than 94% accuracy. The findings for personalized models should be validated with data collected in several independent sessions before personalized models are used for actual patients, as the current results have been achieved with training data from the same session as the test data.

Training a classifier on a training set with both healthy subject and stroke patient data was tested in one of the thesis' final experiments. This did not affect accuracy negatively, but resulted in a slight overall improvement when compared to the results of classifiers targeting each of the groups separately. Making generalized models for larger groups, targeting both healthy subjects and different patient populations, could therefore be a possibility. The results of the same experiment also indicate that using training data from healthy subjects only for some activities could be beneficial when classifying data from stroke patients and vice versa.

Two of the thesis' experiments evaluated semipopulation classifiers as an alternative to personalized classifiers trained with an ordinary random forests algorithm. Semipopulation classifiers did not achieve higher accuracy than regular random forests classifiers with access to subject training data. However, the classifiers' results were impressive given that the sub-models, which make up such classifiers, are trained on training data from only one subject. The final experiment evaluated whether semipopulation classifiers could be used to find similarities between subjects, with the aim of using these similarities to estimate one subject's health based on which subjects it was found to be similar to. The experiment did not find that the similarities were significantly related to physical ability. Semipopulation classifiers could therefore not be used to draw conclusions about a subject's health condition.

7.2 Contributions

This thesis' two research goals were, first, to create a HAR system which performs accurate classification for people with motor impairments (i.e. stroke patients) and, second, to create a system which can be used as a diagnostics tool for people with motor impairments.

The first goal has been fulfilled: The thesis has resulted in a system which is capable of recognizing activities for stroke patients with approximately 94% accuracy using non-personalized models. Models are available for different numbers of sensors and configurations, which makes the system highly adaptable to the needs of different patients. The program is not very resource intensive and could be run on almost any personal computer released in the last decade as long as an implementation of the Python programming language is available for it.

The first research question with regards to making the HAR system was which sensor combinations work best for the recognition task. This was answered by the experiments in sections 6.2 and 6.3. With regards to the second research question, which asked what amount of training data is necessary for adequate classification, the single-personalization semipopulation classifiers in the same experiments showed that it is possible to perform adequate classification for a subject with a classifier trained only on one other subject's training data.

The plan for fulfilling for the second goal was to use similarities found when calibrating semipopulation classifiers to estimate a patient's walking speed. The experiment in section 6.5 showed that this could not be done. The first research question related to this goal asked whether similarities in movement, as recognized by a classifier, would be indicative of the patient's health. As the experiment showed, this was not the case. Consequently, the second related research question could not be answered, as this asked which sensor placements served this purpose best.

For future research, a thorough investigation of how different sensor placements and combinations affect accuracy when recognizing ambulatory movements will probably be the most useful contribution of this thesis. These findings were presented in the experiments in sections 6.2 and 6.3. The most important of these findings were that the lower back and thigh sensors are best for the classification task, and that matching matching the thigh sensors by the body's unaffected and affected sides is better for the classification than matching by left and right sides. No studies were found in the literature search that have performed a similar investigation. The results can be used to choose sensor placements in future data collections from stroke patients and patient groups with similar disabilities.

Another useful finding is that of section 6.4, which showed that a classifier trained on training data from both healthy subjects and stroke patients could result in the same accuracy for both groups as using a classifier trained only on data from their own group. The results in section 6.1 showed that training only on data from healthy subjects was not sufficient to classify data from stroke patients. Together these findings indicate that, while training data from groups with a certain disability is necessary to classify activities in these groups, classifiers do not have to be trained exclusively on training data from these groups. Training data from specific patient groups could possibly just be used to expand the training set for an existing classifier targeting healthy subjects, resulting in a classifier suited for a larger part of the population.

The final contribution which should be mentioned is the Trondheim Chronic Stroke data set. This data set was collected and labeled by Atle Kongsvold and converted and synchronized by this thesis' author. Future research targeting stroke patients will hopefully be able to use this data set to develop novel methods and algorithms that improve activity recognition for stroke patients.

7.3 Future Work

7.3.1 Data Set Improvements

Section 6.2.2 discussed how *bending* was often misclassified as either *standing* or *sitting*. One possible cause of this is that the definition of *bending*, seen in appendix B, includes both bending when sitting down and standing. *Bending* samples should therefore be assigned to two disjoint classes: *bending (sitting)* and *bending (standing)*. Judging from the definition, the activity should always be surrounded by samples of either *sitting* or *standing*. Re-labeling could therefore be done automatically by iterating through the data set and assigning the new label types according to whether *sitting* or *standing* preceded the *bending* instances.

The system was never able to recognize *running* using non-personalized classifiers.

Therefore, the activity should be removed from the data set entirely, unless more data for the activity becomes available. As seen in appendix A, which presents the collection protocol, the activity was suggested to the subjects by the collector, but performing it was optional. Only two subjects ran during their session, as seen in appendix C. This indicates a low likelihood of stroke patients performing this activity in daily life.

The experiment in section 6.4 showed that F1-scores for *bending* and *cycling* improved when a classifier was trained on data from both healthy subjects and stroke patients. Future experiments should evaluate whether adding samples for these two activities is beneficial when training a classifier targeted exclusively at stroke patients.

7.3.2 Validation Outside a Laboratory

Research on new stroke patient treatments and adaptation of current treatments to a patient's needs are two situations where this thesis' HAR system could be useful. Research outcomes and treatment decisions will have direct influence on the health of one or more persons. If data from this system is to influence such decisions, it is important that the findings in this thesis are validated outside of a laboratory.

The findings of the specialization project preceding this thesis, Larsen and Vågeskar [2016], give us reason to exercise caution when using a HAR system trained on data from a laboratory in real-life settings. Section 3.1.8 explained how this project found that the HAR system developed by Hessen and Tessem [2016] performed significantly worse when tested on data gathered outside of a laboratory. The same section presented other examples of research which led to the same conclusion about transitioning from laboratory to non-laboratory data. Some factors could have made the transition harder for Hessen and Tessem's HAR system than for the system presented in this thesis. For example, the activity set recognized by Hessen and Tessem included more activities, such as *stairs (ascending)* and *stairs (descending)*. Given the limited activity set recognized by this thesis' system, it is not necessary that we will see the same decline in performance.

Collecting a new labeled data set should be avoided unless it is considered to be absolutely necessary: The total time associated video recording and labeling is several times longer than the actual duration of a session. A less expensive alternative which could be used to validate this system would be to compare its outputs with the results of an observation for the same time period. An observer would follow a stroke patient wearing sensors for some period of time and note the time spent performing different activities. To make the task easier, the activities could just be those which are most critical to treatment decisions and research conclusions, e.g. *walking* and *standing*. Afterwards, the estimates of the observer and estimates from the HAR system should be compared. Differences within a reasonable margin of error should be tolerated. If the differences are outside these margins of error, a new, non-laboratory data set for stroke patients should be collected.

7.3.3 Expanding to Other Patient Groups

There are other patient groups for which HARs systems can be beneficial. Data has already been collected by employees at the The Faculty of Medicine and Health Sciences for children with cerebral palsy, using the same sensor setup as was used in this thesis.

Other ambulatory HAR systems have targeted patients with Parkinson's disease (for example Dijkstra et al. [2010]), and systems for gait recognition have targeted patients with Huntington's disease (for example Mannini et al. [2016], explained in section 3.2.3).

Section 6.4 showed that a classifier could be trained on data from both healthy subjects and stroke patients (both with only adult subjects) and still perform equally well for both groups as classifiers trained exclusively for these groups. It will be interesting to see whether additional patient groups, possibly from other age ranges, can be added to such a classifier without impacting classification quality. This could open up the possibility of general HAR systems which are able to classify data from larger groups of the population, including both healthy adults and groups of different ages with different disabilities.

7.3.4 Using Time Spent on Activities to Predict Health Condition

In the introduction, it was hypothesized that trends and similarities in time spent on different activities over several weeks could be used to predict a person's future health. These predictions would be based by comparing the information about a current patient with trends from previous patients, extracting information from their medical records.

An example of a trend which will probably be useful to follow is the progression in active time, mainly defined by the time labeled as *walking*, but possibly other activities such as *cycling*. Patients should wear sensors from as early on in the rehabilitation process as possible, and preferably for many weeks consecutively. A rapid decline in a single patient's active time could be used by the patient's physical therapist to start interventions early. Trends in active time for many patients over long periods combined with knowledge about interventions such as exercises, medications, and surgery, could be used to compile large data sets which could be used to select the best treatments.

As 12 000 Norwegians suffer a stroke each year, there are a lot of potential contributors to such a data set. If these users are willing to let data about their health and activities be used in research, it is possible to envision that research on a system could start in a year or two. Such a system would have to draw upon knowledge from the fields of data mining combined with expert knowledge from the medical field.

Appendices

Appendix A

Trondheim Stroke Data Collection Protocol

This is the protocol as received on the 25th of April 2017. With the exception of referencing the table by ID rather than position on the page, only typographic changes have been made.

A.1 Project Title: Activity Detection in Stroke Patients

The aim of the project is to develop an activity detection algorithm that can be used to evaluate physical activity among (chronic) stroke patients. Two systems are proposed for this purpose:

- Wristband recording for long-term follow-up (maybe connected to smartphone app)
- Thigh/low back recording for pre/post evaluation of treatment (i.e., the carry-over effect to activity level during daily life)

A.1.1 Summary of Protocol

Record activity data along with go-pro recording in 12 stroke patients

- 1 hr recording per patient
- **Semi-structured protocol** with the activities in Table A.1 (**minimum 5 min accumulated time** on each activity)
- A sequence with the "Time-up-and-go test" and maximal 10 m walking speed should be included.

Activity	**Min**
Sitting	
Standing	
Moving (“Shuffling” while standing)	
Walking (including up and down stairs	
Biking (stationary bike)	
Lying down	
Optional activities	
Jogging	

Table A.1: Activities

- In addition, it may also be interesting to include a sequence with ‘voluntary/active’ arm movement during ADL[1] tasks (compared to passive arm movements during, e.g., walking)
- Set up with 4 sensors
  - Wristband (both hands)
  - Thigh
  - Lower back
- Perform three heel drops at start and end of recording to synchronize signals.

[1] Activity of Daily Living

Appendix B

Activity Definitions

Table B.1: Activity definitions used when labeling from video

Activity	Definition
Walking	Locomotion towards a destination with one stride or more, (one step with both feet, where one foot is placed at the other side of the other). Walking could occur in all directions. Walking along a curved line is allowed.
Running	Locomotion towards a destination, with at least two steps where both feet leave the ground during each stride. **For chest mounted camera:** Running can be inferred when trunk moves forward in a constant upward-downward motion with at least two steps. Running along a curved line is allowed.
Shuffling	Stepping in place by non-cyclical and non-directional movement of the feet. Includes turning on the spot with feet movement not as part of walking bout. **For chest mounted camera:** Without being able to see the feet, if movement of the upper body and surroundings indicate non-directional feet movement, shuffling can be inferred.
Stairs, ascending or descending	**Start:** Heel-off of the foot that will land on the first step of the stairs. **End:** When the heel-strike of the last foot is placed on flat ground. (If both feet rest on the same step with no feet movement, standing should be inferred.)
Standing	Upright, feet supporting the person's body weight, with no feet movement, otherwise this could be shuffling/walking. Movement of upper body and arms is allowed until forward tilt and arm movement occurs below knee height. Then this should be inferred as bending. **For chest mounted camera:** If feet position is equal before and after upper body movement, standing can be inferred. Without being able to see the feet, if upper body and surroundings indicate no feet movement, standing can be inferred.
Sitting	When the person's buttocks is on the seat of the chair, bed or floor. Sitting can include some movement in the upper body and legs; this should not be tagged as a separate transition. Adjustment of sitting position is allowed.
Lying	The person lies down. Adjustment after lying down is allowed if it does not lead to a change between the prone, supine, right and left lying positons. Movement of arms and head is allowed.
Transition	Transitioning between any of the activities listed here.
Bending	While standing/sitting, bending towards something below knee-height is tagged as bending. Steps can occur during bending.
Picking	Refers to picking/placing/touching an object from below knee height. Picking occurs when the trunk is at its lowest and the person has touched/placed/picked an object. When the person starts to rise the trunk, picking finishes, and bending begins. Adjustment of position while picking is allowed.
Undefined activity	Until all the sensors are attached, or final adjustment made to position the video camera can be tagged as undefined. All postures/movements that can not be clearly identified due to blocking of the camera/view should be tagged as undefined.
Cycling (sitting)	Pedaling while the buttocks is placed at the seat. Cycling starts on first pedaling and finishes when pedaling ends. **Not pedaling:** Sitting without pedaling should be tagged separate as sitting.
Non-vigorous activity	All non-cyclic movements that are recognizable, but do not classify according to the definitions. Can occur in all directions. Can be crawling, rolling, cleaning the floor etc.

Appendix C

Trondheim Stroke Subjects

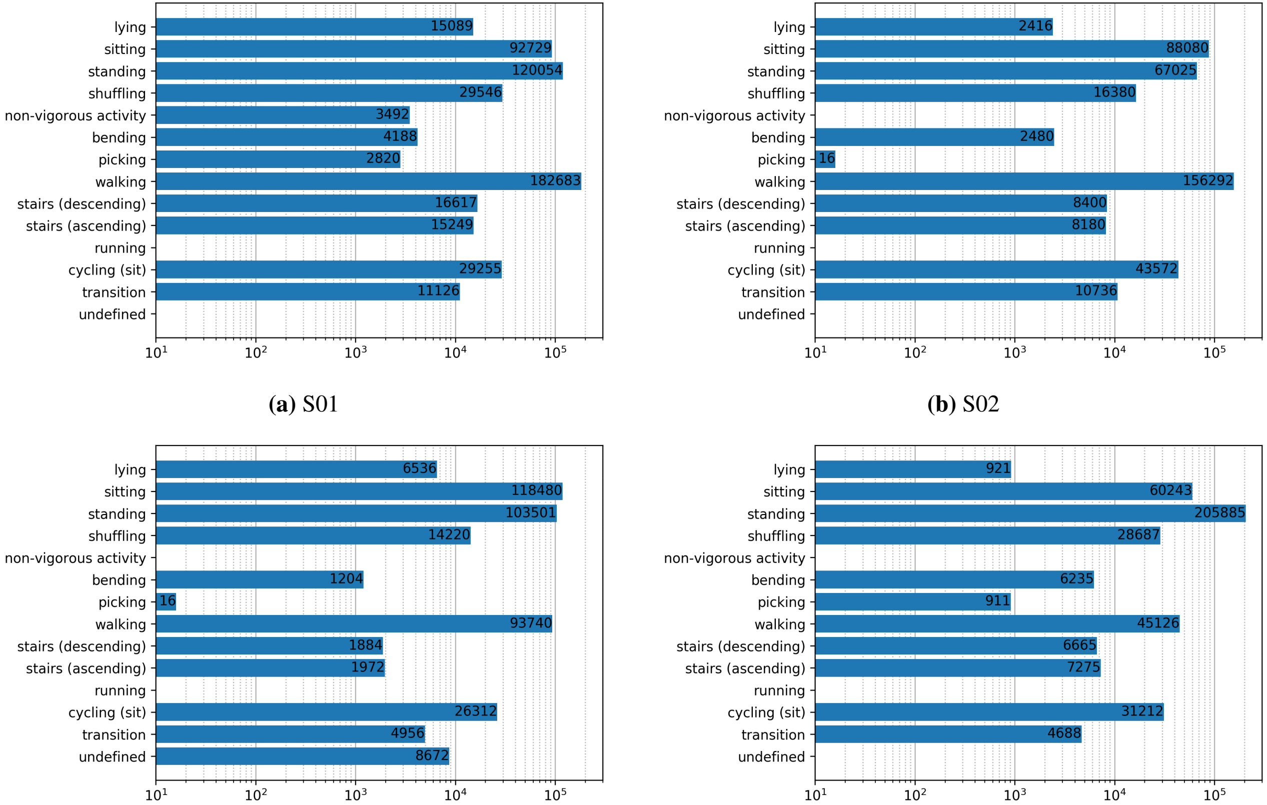
Subject ID	Gender	Affected side	Age (years)	Height (cm)	Weight (kg)	10 meter walk (s)	TUG (s)
S01	F	Left	37	172	67.5	11.18	17.29
S02[1]	F	Left	49	171	68	22.87[2]	25.12
S03	F	Left	65	168	79	10.07	12.56
S05	M	Right	44	186	86	6.71	10.66
S06	M	Right	61	179	95	17.56	20.64
S07	M	Right	51	172	74	11.19	10.39
S08	M	Left	49	180	95	12.32	19.14
S09	M	Left	60	183	84	6.97	9.20
S10	M	Left	72	180	70	12.38	16.45
S11	M	Right	38	171	83	8.31	8.68
S12	F	Left	65	167	?	17.68[3]	28.92
S13	M	Left	68	176	92	5.80	10.89
S14	M	Right	60	180	100	4.68	7.80
S15	F	Left	58	154	52	26.84	28.20
S16	M	Left	53	178	79	13.62	17.94

Table C.1: Subject data for all subjects in the TCS data set

[1] LB sensor is missing. Also performed stair walking very slowly, pausing for many seconds after each step.

[2] Time with crutch. Without crutch: 52.13 s

[3] Time with crutch. Without crutch: 19.56 s



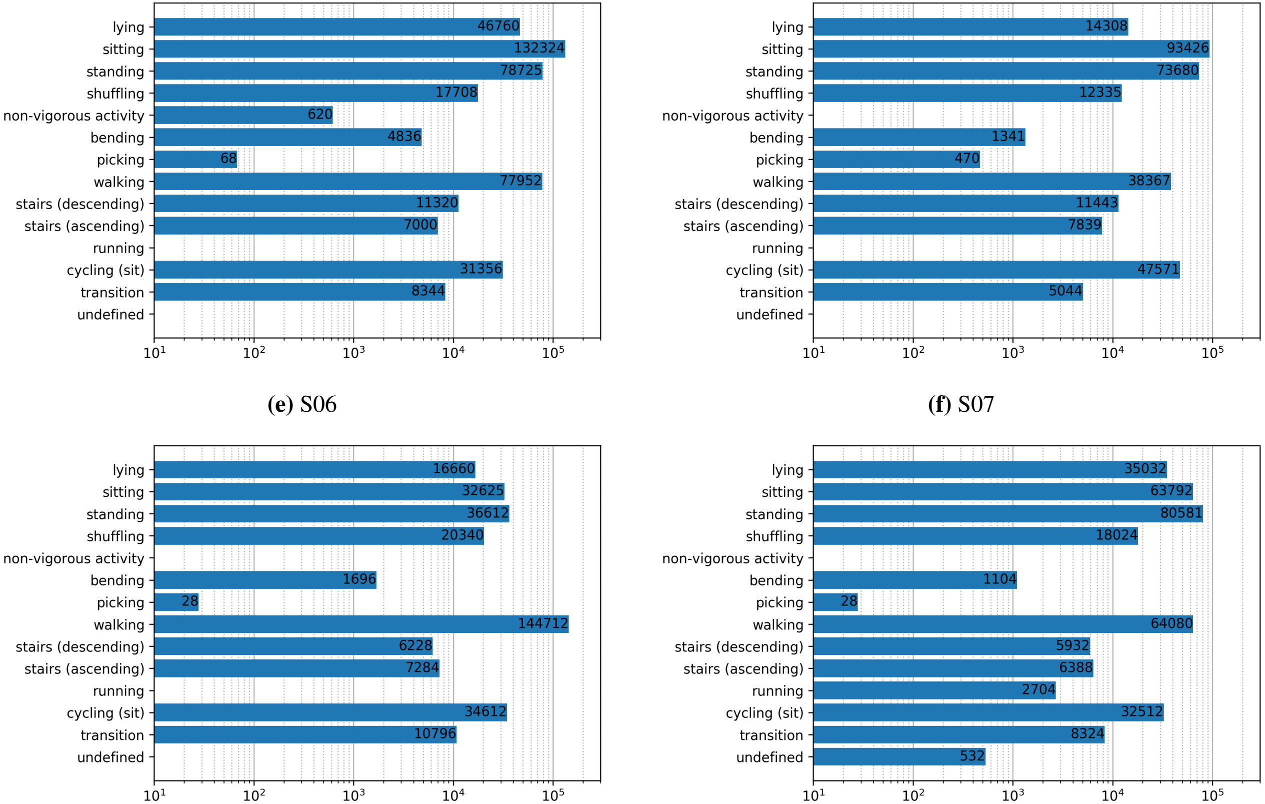
(a) S01

(b) S02

(c) S03

(d) S05

Figure C.1: Activity distribution for each individual in TCS



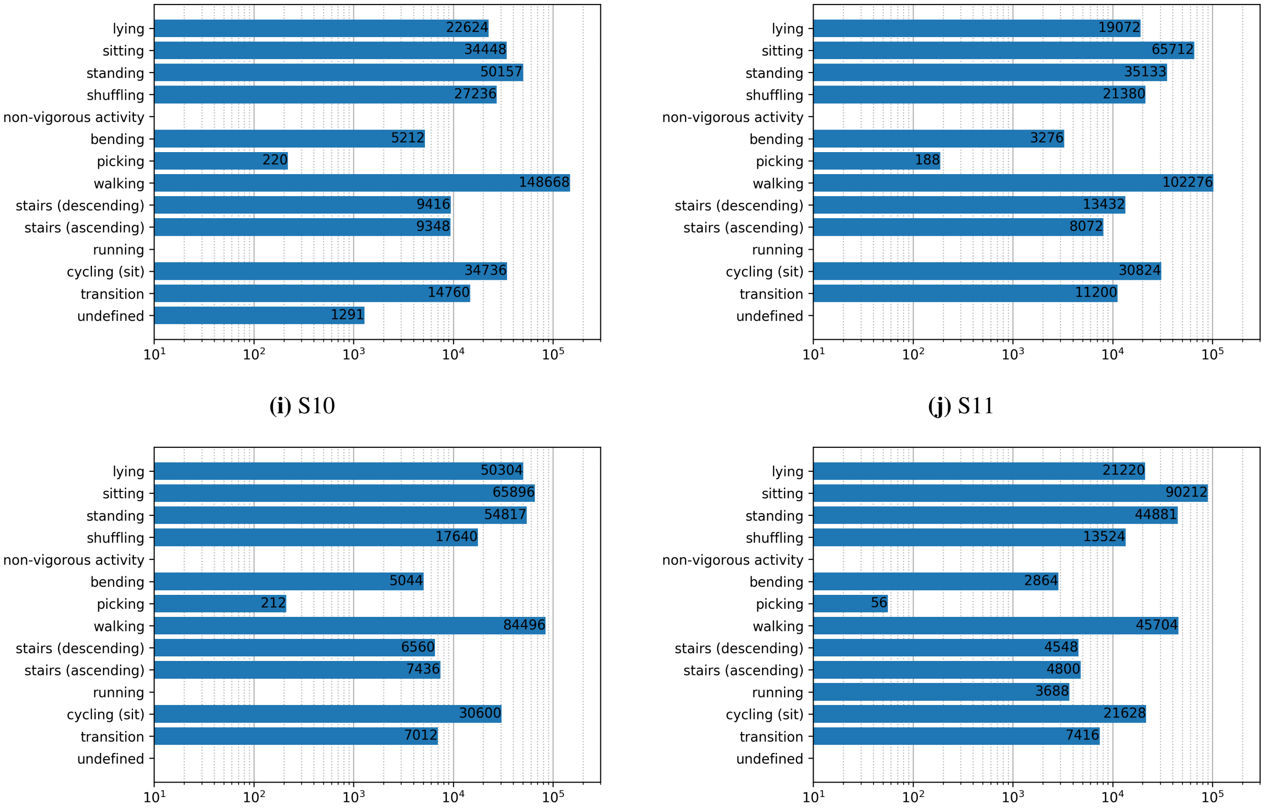
(e) S06

(f) S07

(g) S08

(h) S09

Figure C.1: Activity distribution for each individual in TCS



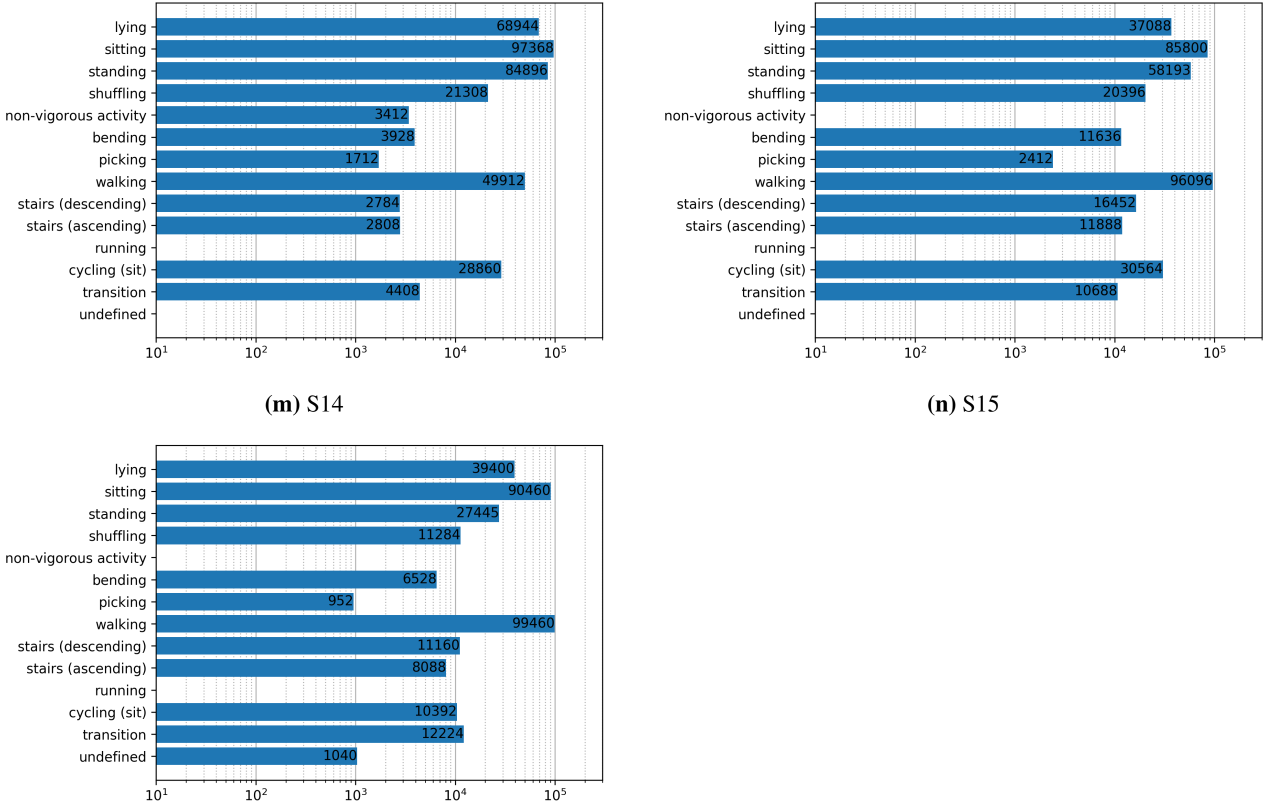
(i) S10

(j) S11

(k) S12

(l) S13

Figure C.1: Activity distribution for each individual in TCS



(m) S14

(n) S15

(o) S16

Figure C.1: Activity distribution for each individual in TCS

Appendix D

Seed Values

In repeated experiments, the following values were used as seeds for the random value functions. At the first iteration, the first value was used; at the second iteration, the second value was used and so on.

The values were drawn by random.org, a service delivering truly random numbers based on atmospheric noise. Probabilities were distributed uniformly among all numbers from 1 up to and including 2^{28}.

1. 2554679
2. 206663454
3. 16273975
4. 262404977
5. 130696134
6. 92839481
7. 32997544
8. 204158098
9. 203423330
10. 76761199

Bibliography

A. Albarbar, A. Badri, J. K. Sinha, and A. Starr. Performance evaluation of MEMS accelerometers. *Measurement: Journal of the International Measurement Confederation*, 42(5):790–795, 2009. ISSN 02632241. doi:10.1016/j.measurement.2008.12.002. URL http://dx.doi.org/10.1016/j.measurement.2008.12.002.

E. K. Antonsson and R. W. Mann. The Frequency Content of Gait. *Journal of Biomechanics*, 18(1):39–47, 1985. doi:10.1016/0021-9290(85)90043-0.

L. Bao and S. S. Intille. Activity Recognition from User-Annotated Acceleration Data. *Pervasive Computing*, pages 1 – 17, 2004. ISSN 03029743. doi:10.1007/b96922.

E. J. Benjamin, M. J. Blaha, S. E. Chiuve, M. Cushman, S. R. Das, R. Deo, S. D. de Ferranti, J. Floyd, M. Fornage, C. Gillespie, C. R. Isasi, M. C. Jiménez, L. C. Jordan, S. E. Judd, D. Lackland, J. H. Lichtman, L. Lisabeth, S. Liu, C. T. Longenecker, R. H. Mackey, K. Matsushita, D. Mozaffarian, M. E. Mussolino, K. Nasir, R. W. Neumar, L. Palaniappan, D. K. Pandey, R. R. Thiagarajan, M. J. Reeves, M. Ritchey, C. J. Rodriguez, G. A. Roth, W. D. Rosamond, C. Sasson, A. Towfighi, C. W. Tsao, M. B. Turner, S. S. Virani, J. H. Voeks, J. Z. Willey, J. T. Wilkins, J. H. Wu, H. M. Alger, S. S. Wong, and P. Muntner. Heart Disease and Stroke Statistics—2017 Update: A Report From the American Heart Association. *Circulation*, 135(10):e146–e603, 2017. ISSN 15244539. doi:10.1161/CIR.0000000000000485.

A. Bhattacharya, E. P. McCutheon, E. Shvartz, and J. E. Greenleaf. Body acceleration distribution and O2 uptake in humans during running and jumping. *Journal of Applied Physiology*, 49:881–887, 1980.

H. Blockeel. Observation Language. In *Encyclopedia of Machine Learning*, pages 733–735. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi:10.1007/978-0-387-30164-8_608. URL http://dx.doi.org/10.1007/978-0-387-30164-8_608.

H. Blockeel. Hypothesis Language. In *Encyclopedia of Machine Learning and Data Mining*, pages 625–629. Springer US, Boston, MA, 2017a. ISBN 978-1-4899-7687-1.

doi:10.1007/978-1-4899-7687-1_372. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_372.

H. Blockeel. Hypothesis Space. In *Encyclopedia of Machine Learning and Data Mining*, pages 629–632. Springer US, Boston, MA, 2017b. ISBN 978-1-4899-7687-1. doi:10.1007/978-1-4899-7687-1_373. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_373.

R. W. Bohannon. Comfortable and maximum walking speed of adults aged 20-79 years: Reference values and determinants. *Age and Ageing*, 26(1):15–19, 1997. ISSN 00020729. doi:10.1093/ageing/26.1.15.

R. W. Bohannon. Reference values for the timed up and go test: a descriptive meta-analysis. *Journal of geriatric physical therapy (2001)*, 29(2):64–8, 2006. ISSN 1539-8412. doi:10.1519/00139143-200608000-00004. URL http://www.ncbi.nlm.nih.gov/pubmed/16914068.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 08856125. doi:10.1023/A:1010933404324.

A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 1(June):1–33, 2014. ISSN 03600300. doi:http://dx.doi.org/10.1145/2499621. URL http://dl.acm.org/citation.cfm?doid=2578702.2499621%5Cnhttp://dl.acm.org/citation.cfm?id=2499621.

N. A. Capela, E. D. Lemaire, and N. Baddour. Feature Selection for Wearable Smartphone- Based Human Activity Recognition with Able bodied , Elderly , and Stroke Patients. *PLoS ONE*, 10(4):1–18, 2015. doi:10.1371/journal.pone.0124414.

N. A. Capela, E. D. Lemaire, N. Baddour, M. Rudolf, N. Goljar, and H. Burger. Evaluation of a smartphone human activity recognition application with able-bodied and stroke participants. *Journal of Neuroengineering and Rehabilitation*, 13(1):5, 2016. ISSN 1743-0003. doi:10.1186/s12984-016-0114-0. URL http://www.jneuroengrehab.com/content/13/1/5%5Cnhttp://www.scopus.com/inward/record.url?eid=2-s2.0-84955251662&partnerID=tZOtx3y1.

V. H. Cheung, L. Gray, and M. Karunanithi. Review of accelerometry for determining daily activity among elderly patients. *Archives of Physical Medicine and Rehabilitation*, 92(6):998–1014, 2011. ISSN 00039993. doi:10.1016/j.apmr.2010.12.040. URL http://dx.doi.org/10.1016/j.apmr.2010.12.040.

I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay. Optimal placement of accelerometers for the detection of everyday activities. *Sensors (Basel, Switzerland)*, 13(7):9183–9200, 2013. ISSN 14248220. doi:10.3390/s130709183.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts; London, England, 3rd edition, 2007. ISBN 978-0-262-03384-8.

T. G. Dietterich and E. B. Kong. Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithms. *Machine Learning*, 255:0–13, 1995. doi:http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.6820. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.2702&amp;rep=rep1&amp;type=pdf.

B. Dijkstra, Y. P. Kamsma, and W. Zijlstra. Detection of gait and postures using a miniaturized triaxial accelerometer-based system: Accuracy in patients with mild to moderate Parkinson's disease. *Archives of Physical Medicine and Rehabilitation*, 91 (8):1272–1277, 2010. ISSN 00039993. doi:10.1016/j.apmr.2010.05.004. URL http://dx.doi.org/10.1016/j.apmr.2010.05.004.

B. H. Dobkin, X. Xu, M. Batalin, S. Thomas, and W. Kaiser. Reliability and validity of bilateral ankle accelerometer algorithms for activity recognition and walking speed after stroke. *Stroke*, 42(8):2246–2250, 2011. ISSN 00392499. doi:10.1161/STROKEAHA.110.611095.

C. Elkan. The Foundations of Cost-Sensitive Learning. *Proceedings of International Joint Conference on Artificial Intelligence*, pages 973–978, 2001. ISSN 10450823. doi:doi=10.1.1.29.514.

F. Foerster, M. Smeja, and J. Fahrenberg. Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in Human Behavior*, 15(5):571–583, 1999. ISSN 07475632. doi:10.1016/S0747-5632(99)00037-0.

J. Fürnkranz. Decision Tree. In *Encyclopedia of Machine Learning and Data Mining*, pages 330–335. Springer US, Boston, MA, 2017. ISBN 978-1-4899-7687-1. doi:10.1007/978-1-4899-7687-1_66. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_66.

S. Geman, E. Bienenstock, and R. Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1–58, 1992. ISSN 0899-7667. doi:10.1162/neco.1992.4.1.1. URL http://www.mitpressjournals.org/doi/10.1162/neco.1992.4.1.1.

C. Globas, C. Becker, J. Cerny, J. M. Lam, U. Lindemann, L. W. Forrester, R. F. Macko, and A. R. Luft. Chronic Stroke Survivors Benefit From High-Intensity Aerobic Treadmill Exercise : A Randomized Controlled Trial. *Neurorehabilitation and Neural Repair*, 26(1):85–95, 2012. doi:10.1177/1545968311418675.

Google. SensorEvent | Android Developers, 2014. URL https://developer.android.com/reference/android/hardware/SensorEvent.html#values.

T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu. epSICAR: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. *7th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2009*, 2009. doi:10.1109/PERCOM.2009.4912776.

J. Guo, X. Zhou, Y. Sun, G. Ping, G. Zhao, and Z. Li. Smartphone-Based Patients' Activity Recognition by Using a Self-Learning Scheme for Medical Monitoring. *Journal of Medical Systems*, 40(6), 2016. ISSN 1573689X. doi:10.1007/s10916-016-0497-2. URL http://dx.doi.org/10.1007/s10916-016-0497-2.

R. Helaoui, M. Niepert, and H. Stuckenschmidt. Recognizing interleaved and concurrent activities: A statistical-relational approach. *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on. IEEE*, pages 1–9, 2011. doi:10.1109/PERCOM.2011.5767586. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5767586.

H.-O. Hessen and A. J. Tessem. Human Activity Recognition With Two Body-Worn Accelerometer Sensors. Master's thesis, Norwegian University of Science and Technology, Trondheim, 2016.

J.-H. Hong, J. Ramos, and A. K. Dey. Toward Personalized Activity Recognition Systems With a Semipopulation Approach. *IEEE Transactions on Human-Machine Systems*, 46(1):101–112, 2016.

T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp '08)*, pages 10–19, 2008. doi:10.1145/1409635.1409638. URL http://portal.acm.org/citation.cfm?doid=1409635.1409638.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*. Springer US, 2013. ISBN 9780387781884. doi:10.1016/j.peva.2007.06.006. URL http://books.google.com/books?id=9tv0taI8l6YC.

H. S. Jørgensen, H. Nakayama, H. O. Raaschou, and T. S. Olsen. Recovery of Walking Function in Stroke Patients : The Copenhagen Stroke Study. *Archives of Physical Medicine and Rehabilitation*, 76(1):27–32, 1995.

S. Kakurai and M. Akai. Clinical experiences with a convertible thermoplastic knee-ankle-foot orthosis for post-stroke hemiplegic patients. *Prosthetics and Orthotics International*, 20(3):191–194, 1996. URL http://journals.sagepub.com/doi/abs/10.3109/03093649609164442.

A. M. Khan, M. H. a. Siddiqi, and S. W. Lee. Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones. *Sensors (Basel, Switzerland)*, 13(10):13099–13122, 2013. ISSN 14248220. doi:10.3390/s131013099.

A. M. Khan, A. Tufail, A. M. Khattak, and T. H. Laine. Activity recognition on smartphones via sensor-fusion and KDA-based SVMs. *International Journal of Distributed Sensor Networks*, 2014, 2014. ISSN 15501477. doi:10.1155/2014/503291.

O. D. Lara and M. A. Labrador. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013. ISSN 1553-877X. doi:10.1109/SURV.2012.110112.00192. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6365160.

O. D. Lara, A. J. Prez, M. A. Labrador, and J. D. Posada. Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and Mobile Computing*, 8(5):717–729, 2012. ISSN 15741192. doi:10.1016/j.pmcj.2011.06.004. URL http://dx.doi.org/10.1016/j.pmcj.2011.06.004.

F. G. Larsen and E. Vågeskar. Investigating the Performance of a Human Activity Recognition System on Out-of-Lab Data. Technical report, The Norwegian University of Science and Technology, Trondheim, 2016.

H.-Y. Lau, K.-Y. Tong, and H. Zhu. Support vector machine for classification of walking conditions of persons after stroke with dropped foot. *Human Movement Science*, 28: 504–514, 2009. doi:10.1016/j.humov.2008.12.003.

J. Law and R. Rennie. Vector. In *Oxford Dictionary of Physics*. Oxford University Press, 2015. ISBN 9780198714743. URL //www.oxfordreference.com/10.1093/acref/9780198714743.001.0001/acref-9780198714743-e-3222.

L. Liao. *Location-Based Activity Recognition*. PhD thesis, University of Washington, 2006.

C. X. Ling and V. S. Sheng. Cost-Sensitive Learning. In *Encyclopedia of Machine Learning and Data Mining*, pages 285–289. Springer US, Boston, MA, 2017. ISBN 978-1-4899-7687-1. doi:10.1007/978-1-4899-7687-1_181. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_181.

K. Liu, Y. Wang, R. Chen, T. Chu, and J. Bi. A Survey of Human Activity Recognition Using Smartphones. *Journal of Residuals Science & Technology*, 13(8):1–10, 2016. doi:10.12783/issn.1544-8053/13/8/385.

S. Lomax and S. Vadera. A survey of cost-sensitive decision tree induction algorithms. *ACM Computing Surveys*, 45(2):1–35, 2013. ISSN 03600300. doi:10.1145/2431211.2431215. URL http://dl.acm.org/citation.cfm?doid=2431211.2431215.

L. Lonini, A. Gupta, K. Kording, and A. Jayaraman. Activity Recognition in Patients with Lower Limb Impairments : Do we need training data from each patient? In *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3265–3268, 2016. ISBN 9781457702204.

R. F. Macko, F. M. Ivey, L. W. Forrester, D. Hanley, J. D. Sorkin, L. I. Katzel, K. H. Silver, and A. P. Goldberg. Treadmill exercise rehabilitation improves ambulatory function and cardiovascular fitness in patients with chronic stroke: A randomized, controlled trial. *Stroke*, 36(10):2206–2211, 2005. ISSN 00392499. doi:10.1161/01.STR.0000181076.91805.89.

T. Maekawa and Y. Kishino. Activity recognition with hand-worn magnetic sensors. *Personal and Ubiquitous Computing*, 17(6):1085–1094, 2013. doi:10.1007/s00779-012-0556-8.

A. Mannini, D. Trojaniello, A. Cereatti, and A. M. Sabatini. A machine learning framework for gait classification using inertial sensors: Application to elderly, post-stroke and huntington's disease patients. *Sensors (Switzerland)*, 16(1), 2016. ISSN 14248220. doi:10.3390/s16010134.

U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pages 4–7, 2006. doi:10.1109/BSN.2006.6.

J. Mehrholz, M. Pohl, and B. Elsner. Treadmill training and body weight support for walking after stroke (Review). *Cochrane Database of Systematic Reviews*, pages 1–88, 2014. ISSN 1469-493X. doi:10.1002/14651858.CD002840.pub3.

T. M. Mitchell. *Machine Learning*. WCB/McGraw-Hill, 1 edition, 1997. ISBN 9780070428072.

S. E. Nadeau, S. S. Wu, B. H. Dobkin, S. P. Azen, D. K. Rose, J. K. Tilson, S. Y. Cen, and P. W. Duncan. Effects of Task-Specific and Impairment-Based Training Compared With Usual Care on Functional Walking Ability After Inpatient Stroke Rehabilitation. *Neurorehabilitation and Neural Repair*, 27(4):370–380, 2013. ISSN 1545-9683. doi:10.1177/1545968313481284. URL http://journals.sagepub.com/doi/10.1177/1545968313481284.

NINDS. Stroke: Hope Through Research | National Institute of Neurological Disorders and Stroke, 1999. URL https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Hope-Through-Research/Stroke-Hope-Through-Research.

Norsk Helseinformatikk. Hjerneslag, 2017. URL https://nhi.no/sykdommer/hjernenervesystem/hjerneslag-og-blodninger/hjerneslag/.

R. T. Olszewski, R. Maxion, D. Siewiorek, and D. Banks. Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy Thesis Committee : Generalized Feature Extraction for Structural Pattern Reco. Technical Report February, School of Computer Science, Carnegie Mellon University, Pittsburgh, 2001.

N. Pannurat, S. Thiemjarus, E. Nantajeewarawat, and I. Anantavraslip. Analysis of Optimal Sensor Positions for Activity Classification and Application on a Different Data Collection Scenario. *Sensors*, 17(March), 2017. ISSN 14248220. doi:10.20944/preprints201703.0122.v1.

L. Paul, S. Brewster, S. Wyke, J. M. R. Gill, G. Alexander, A. Dybus, and D. Rafferty. Physical activity profiles and sedentary behaviour in people following stroke: a cross-sectional study. *Disability and Rehabilitation*, 38(4):362–367, 2016. ISSN 1464-5165. doi:10.3109/09638288.2015.1041615. URL http://www.tandfonline.com/action/journalInformation?journalCode=idre20http://informahealthcare.com/dre.

J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986. ISSN 15730565. doi:10.1023/A:1022643204877.

S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks*, 6 (2):1–27, 2010. ISSN 15504859. doi:10.1145/1689239.1689243. URL http://portal.acm.org/citation.cfm?doid=1689239.1689243.

D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. Del R. Millàn. Collecting complex activity datasets in highly rich networked sensor environments. *INSS 2010 - 7th International Conference on Networked Sensing Systems*, pages 233–240, 2010. doi:10.1109/INSS.2010.5573462.

S. H. Roy, M. S. Cheng, S. S. Chang, J. Moore, G. De Luca, S. H. Nawab, and C. J. De Luca. A Combined sEMG and Accelerometer System for Monitoring Functional Activity in Stroke. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(6):585–594, 2009. ISSN 15344320. doi:10.1109/TNSRE.2009.2036615.

J. Rueterbories, E. G. Spaich, B. Larsen, and O. K. Andersen. Methods for gait event detection and analysis in ambulatory systems. *Medical Engineering and Physics*, 32 (6):545–552, 2010. ISSN 13504533. doi:10.1016/j.medengphy.2010.03.007. URL http://dx.doi.org/10.1016/j.medengphy.2010.03.007.

S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, Inc., Upper Saddle River, NJ, 3 edition, 2010. ISBN 9780136042594.

C. Sammut and G. I. Webb. Accuracy. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*, pages 9–10. Springer US, Boston, MA, 2010a. ISBN 978-0-387-30164-8. doi:10.1007/978-0-387-30164-8_3. URL http://dx.doi.org/10.1007/978-0-387-30164-8_3.

C. Sammut and G. I. Webb. F 1-Measure. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*, page 397. Springer US, Boston, MA, 2010b. ISBN 978-0-387-30164-8. doi:10.1007/978-0-387-30164-8_298. URL http://dx.doi.org/10.1007/978-0-387-30164-8_298.

C. Sammut and G. I. Webb. Supervised Learning. In *Encyclopedia of Machine Learning and Data Mining*, pages 1213–1214. Springer US, Boston, MA, 2017a. ISBN 978-1-4899-7687-1. doi:10.1007/978-1-4899-7687-1_803. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_803.

C. Sammut and G. I. Webb. Cross-Validation, 2017b. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_190.

C. Sammut and G. I. Webb. Data Set. In *Encyclopedia of Machine Learning and Data Mining*, page 327. Springer US, Boston, MA, 2017c. ISBN 978-1-4899-7687-1.

doi:10.1007/978-1-4899-7687-1_196. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_196.

C. Sammut and G. I. Webb. Inductive Bias. In *Encyclopedia of Machine Learning and Data Mining*, page 641. Springer US, Boston, MA, 2017d. ISBN 978-1-4899-7687-1. doi:10.1007/978-1-4899-7687-1_390. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_390.

C. Sammut and G. I. Webb. Specificity. In *Encyclopedia of Machine Learning and Data Mining*, page 1167. Springer US, Boston, MA, 2017e. ISBN 978-1-4899-7687-1. doi:10.1007/978-1-4899-7687-1_770. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_770.

A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3):535–554, 1959. ISSN 0018-8646. doi:10.1147/rd.33.0210.

Scikit-Learn Developers. 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.18.1 documentation, 2016. URL http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

C. E. Shannon. Communication in the Presence of Noise. *Proceedings of The IEEE*, 3(2): 1–11, 1998. ISSN 00968390. doi:10.1109/JRPROC.1949.232969. URL papers://b601f53a-84e1-4f0c-abd7-620bc101dbfb/Paper/p4431.

E. M. Tapia, S. S. Intille, W. Haskell, K. W. J. Larson, A. King, and R. Friedman. Real-Time Recognition of Physical Activities and their Intensitiies Using Wireless Accelerometers and a Heart Monitor. *International Symposium on Wearable Computers*, pages 37–40, 2007. ISSN 15504816. doi:10.1109/ISWC.2007.4373774.

K. M. Ting. Precision and Recall. In *Encyclopedia of Machine Learning*, page 781. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi:10.1007/978-0-387-30164-8_652. URL http://dx.doi.org/10.1007/978-0-387-30164-8_652.

P. H. Veltink, H. B. J. Bussmann, W. de Vries, W. L. J. Martens, and R. C. Van Lummel. Detection of static and dynamic activities using uniaxial\naccelerometers. *IEEE Transactions on Rehabilitation Engineering*, 4(4):375–385, 1996. ISSN 1063-6528. doi:10.1109/86.547939. URL http://ieeexplore.ieee.org/ielx4/86/11962/00547939.pdf?tp=&arnumber=547939&isnumber=11962.

G. Weiss, K. McCarthy, and B. Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? In *Proceedings of the 2007 International Conference on Data Mining*, pages 1–7, Las Vegas, NV, 2007. CSREA Press. URL http://storm.cis.fordham.edu/~gweiss/papers/dmin07-weiss.pdf.

N. Yazdi, F. Ayazi, and K. Najafi. Micromachined inertial sensors. *Proceedings of the IEEE*, 86(8):1640–1658, 1998. ISSN 00189219. doi:10.1109/5.704269.

J. Ye, G. Stevenson, and S. Dobson. KCAR: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing*, 19:47–70, 2015. doi:10.1016/j.pmcj.2014.02.003.

X. Zhu. Semi-supervised Learning. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 1142–1147. Springer US, Boston, MA, 2017. ISBN 978-1-4899-7687-1. doi:10.1007/978-1-4899-7687-1_749. URL http://dx.doi.org/10.1007/978-1-4899-7687-1_749.