UiO : Department of Informatics
University of Oslo

Detecting Human Emotions Using Smartphone Accelerometer Data

Andreas Færøvig Olsen

Master's Thesis Autumn 2016

Detecting Human Emotions Using Smartphone Accelerometer Data

Andreas Færøvig Olsen

15th August 2016

Abstract

The work presented in this thesis concerns the classification of emotions based on smartphone accelerometer data. The data is collected from individuals who have been carrying their phone in a pocket while walking. While other methods of emotion detection generally are more intrusive or only give estimations over longer periods of time, using accelerometer data presents an opportunity for real-time and non-intrusive emotion detection.

An Android app was developed in order to monitor the smartphone accelerometer of the individuals who participated in the study and occasionally request them to submit their emotional state. This way, data is collected from a natural environment rather than a laboratory setting. The recorded data is then processed and used to train different classifiers to be compared. The machine learning algorithms *decision tree*, *support vector machine* and *multilayer perceptron* are used for this purpose.

Emotions are classified in two dimensions: pleasantness and arousal (activation). While the recognition rate for the arousal dimension is promising at 75%, pleasantness is harder to predict, with a recognition rate of 51%. These findings indicate that by only analyzing accelerometer data recorded from a smartphone, it is possible to make predictions of a person's state of activation. Such predictions can be used, especially in conjunction with other methods of emotion detection, to adapt services to the user's emotional state.

Contents

List of Figures

List of Tables

Preface

Chronologically speaking, this is the final page of the thesis. The sun is shining, and my desire to rise from the chair is higher than my ambition to write the story of how this thesis was born. So it goes.

Thank you . . .

AK, for being generous with your support, praise and encouragement and enlightening on the field of psychology.

Jim, for being the kind of supervisor who points me in the right direction and has the gift of tracking down all kinds of errors, factual as well as grammatical.

everyone who helped collect data, provided input on drafts, developed the frameworks used in the project, asked or answered one of the countless questions on Stack Overflow I had to consult in order to make things work, or in any other way contributed to the completion of the thesis.

Andreas Færøvig Olsen
15th August 2016

Chapter 1

Introduction

Is your phone able to detect your emotions?

Our immediate answer to this question is definitely negative. Phones, or computers in general, are not equipped with "emotion detectors". We also don't necessarily *want* the computer to know how we are feeling, because of privacy issues. However, scientific advances in artificial intelligence also bring an interest in letting emotions into the world of computing. Rosalind J. Picard writes in her book *Affective computing* [49]:

> *I have come to the conclusion that if we want computers to be genuinely intelligent, to adapt to us, and to interact naturally with us, then they will need the ability to recognize and express emotions [...]*

A major part of the field of affective computing consists of recognizing human emotions based on one or more types of data collected from the person whose affective state is being evaluated. This is not a question of perfect recognition of emotions, but rather a better-than-random suggestion. The prediction made by the computer can then be used to influence the behaviour of a system that interacts with the human, as in Figure 1.1. This figure, along with all following figures, are made according to the author's understanding of the illustrated concepts, unless otherwise specified.

Computers use many different ways to sense emotions, including, but not limited to physiological measurements, self-reports, phone usage, speech and facial expressions [32]. Common for most of these methods is that they either require some kind of sensor connected to the body, manual input from the user or another obtrusive[1] method of data collection in order to work, or they require the collection of data over a long period of time in order to predict the long-term mood of the user.

While the accuracy of such methods is frequently shown to be high, they do not allow for an instant evaluation of a person's emotions without the use of any external equipment. That means that the practical use is

[1] *Obtrusive* as in requiring a person to be subject to some kind of perceivable interference for the sake of gathering information about his/her emotions.

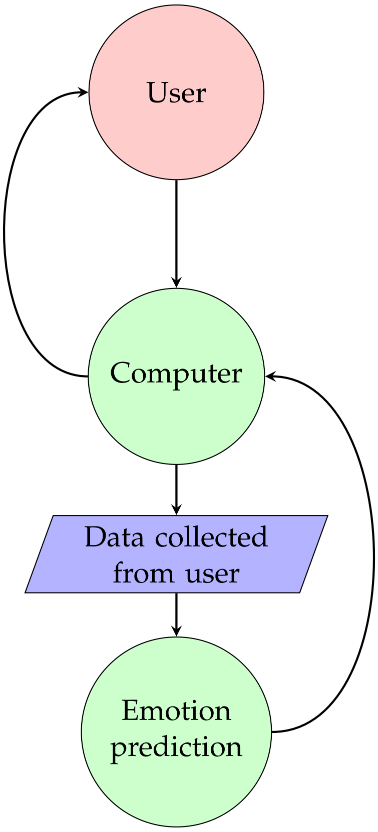


Figure 1.1: Affective computing can be used to influence the interaction between and computer.

somewhat limited, and the methods cannot immediately be used by the general population. In this thesis I will address these issues by introducing a system for predicting a person's short-term emotions solely based on data from the accelerometer embedded in the person's smartphone.

Do we actually move in a different way when we are happy as opposed to when we are sad? Does it make any difference whether we are tired or energetic? Various studies indicate that such a relation might exist, both for walking and other movements [5, 15, 23, 34, 43, 44]. The accelerometer allows us to capture some of this information that connects movement to emotions.

Smartphones have become more and more computationally powerful, and they are now able to perform a high number of computations in addition to just enabling us to perform simple tasks like making phone calls and sending text messages. This gives us the opportunity to record and analyze the accelerometer data without disrupting the normal use of the phone. Using machine learning for classification on user data collected through the app *Emotions*, developed for this work and presented in section 3.2.3, I propose a way to make predictions about the emotions experienced by the user while walking, carrying the phone in a pocket.

1.1 Summary of the work

Asking whether the phone can detect emotions necessitates answering the more elementary question of how to define the concept of *emotions*. Therefore the first course of action was reviewing emotion theory and finding a model that was likely to be suitable for this context.

Having decided to use a dimensionality-based model of emotions, the next problem was how to collect data. The natural way to do this was to create an app that records data at appropriate times from the people who participate in the study. Such an experiment can be conducted in two different ways:

1. In a controlled setting where the participants receive a phone prepared for recording. The participants could e.g. be instructed to carry the phone in a pocket while walking naturally for a set period of time. *Mood induction*[2] techniques could be used in order to encourage a wide selection of emotions being experienced and recorded.

2. Through the use of an app installed on the participants' personal phones. That way the data will be recorded more unobtrusively, and the participants can input the emotional state they are experiencing at the time of recording.

The latter approach was chosen because of the potential for higher data quality when recorded in the participants' natural environment. However, this also required significantly more work in app development and data preprocessing, and the resulting data collection app *Emotions* is described in section 3.2.3

Figure 1.2 shows the different parts of this work. In order to decide more specifically how to handle the various stages of the work, many related studies were reviewed, see section 2.6. They provided input on models of emotion, data collection and everything related to classification. It was decided to compare three different classifiers, using various features extracted from the data set.

When the app was more or less finished, it was tested as data was collected from a few initial participants. Using only a couple of features related to the energy of the movement, the classification showed promising results, but more data and better features were clearly required. This, together with tuning the classifiers, later improved the results of the classification.

The main contribution of this work is the non-obtrusive way in which data is collected. While the classification will not reach the same accuracy as most other comparable methods of emotion detection, it requires no equipment other than a phone that is carried in a pocket while walking. This also makes it a simple addition that can potentially improve the accuracy of more pervasive emotion detection systems. For this purpose,

[2]Manipulating someone through e.g. video clips or social interaction to enter a specific emotional state

an Android app together with a server were developed for data collection, and different features and classifiers have been evaluated.

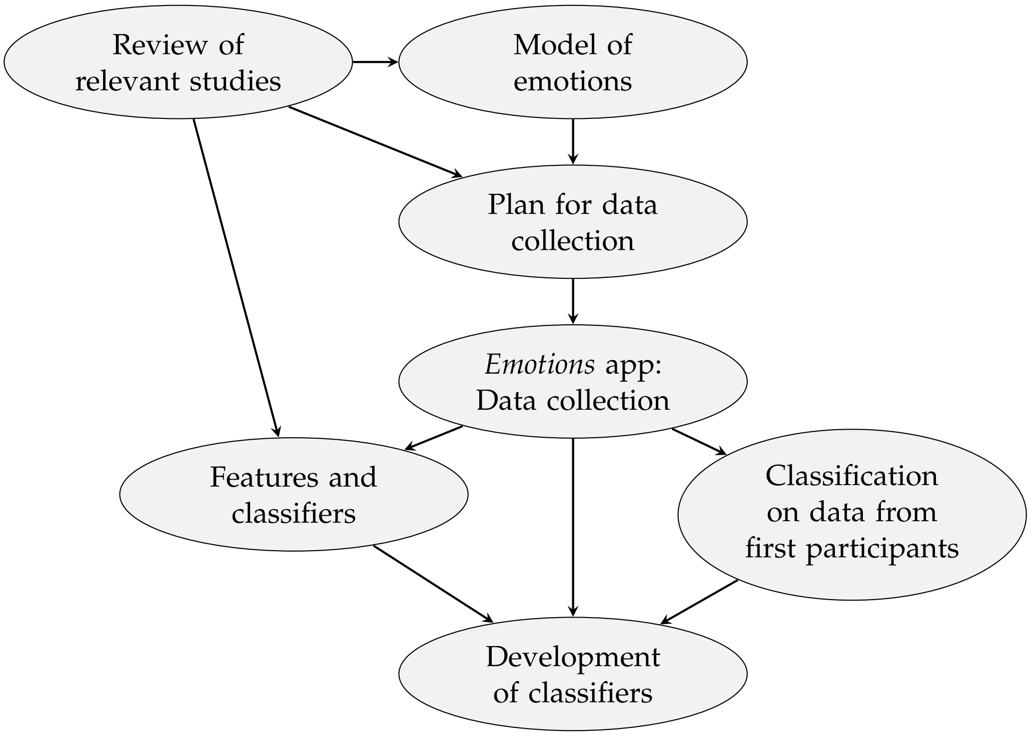


Figure 1.2: The process of designing a system for classifying emotions.

1.2 Applications of emotion recognition

While emotion recognition is an interesting concept in itself, it has many potential applications. Because of the inherent uncertainty involved, the results of this work will likely not be used alone, but rather as part of a collection of different ways to recognize emotions. Below are some interesting areas of application, where the use of an accelerometer could reasonably be included.

Mental health

A multitude of applications designed to promote mental health have been developed [10]. For such applications it could be useful to have additional non-obtrusive ways of detecting the emotional state of the user.

More important are possibly systems designed for use in a medical context. One example of such a system is *Empath*[3] [17], where a patient is monitored over time for symptoms of depression through a range of sensors. This includes monitoring of speech, weight, movement and sleep,

[3]Emotional Monitoring for PATHology

and the data is analysed in order to generate a measurement of the patient's likely level of depression.

A study conducted by Jaques et al. [29, 30] amongst students at MIT aimed to identify which students were at higher risk of depression. Using various types of sensor and smartphone data, including the accelerometer (although not directly as a way to detect emotions), the researchers tried to classify the students' daily mood labeled with self-reported emotional states.

A general summary, as I understand it, of how accelerometer-based emotion detection can be applied for mental health purposes is illustrated in Figure 1.3.

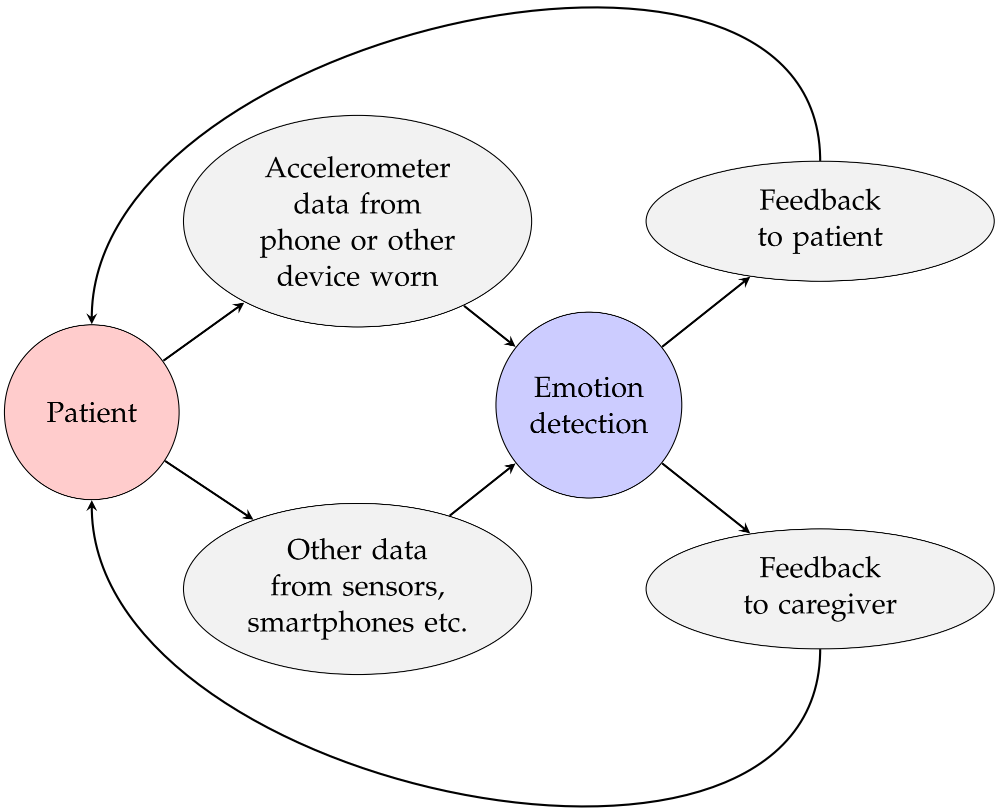


Figure 1.3: Emotion detection in the context of monitoring mental health

Personal assistant

A change in mood is typically accompanied by a change in preferences. Such information could be used by electronic personal assistants like Apple's *Siri* [66]. Over a period of time, the personal assistant could learn that the user tends to request certain actions when he or she is in a certain emotional state. This can then influence how the personal assistant responds and perhaps suggest actions or parameters to the requested actions. If the user e.g. frequently asks the personal assistant to book a table at a restaurant, and the choice of restaurant proves to be dependent on the user's mood, a default parameter (choice of restaurant) can then be set once the likely mood is determined.

Playing music

Music is known to have an impact on emotions [31]. We can also imagine a connection in the other direction – your emotions can decide which music to play. Research has been conducted on how music can help improve a person's mood [28]. Such an *affective music player* can be combined with emotion detection in order to decide at which points in time the music player should interfere.

Computer games

Technological developments are constantly advancing the gaming industry. More detailed graphics, better physical simulations and higher accuracy of controls are some advances that keep making games more immersive. It is easy to imagine that projecting emotions into a game would be interesting both for developers and for gamers. Describing a game which has evolved into a substitution for normal human interaction, the author Ernest Cline says in his 2011 novel *Ready Player One* [12]:

> *Todd13 scowled and his face actually turned red – a sign that he hadn't bothered to turn off his account's real-time emotion feature, which made your avatar mirror your facial expressions and body language.*

The seemingly most popular game during summer 2016, *Pokémon Go* [50], a mobile game where players walk around collecting so-called *Pokémons*, would be a good candidate for implementation of emotion detection based on accelerometer readings, due to the requirement of players walking around in order to play the game. The player's emotions could in some way impact how the game works.

1.3 Chapter overview

The remainder of this thesis is divided into the following chapters:

Chapter 2: Background

Different ways to model emotions are explained here. Additionally, relevant information about smartphones and classification is presented. Finally, there is a review of other studies relevant to this work.

Chapter 3: Detecting Emotions

This chapter contains information about data collection, preprocessing and feature extraction.

Chapter 4: Experiments

All experiments and results are found here.

Chapter 5: Conclusion and Further Work

A conclusion of the results, in addition to some ideas for improvement and future extension of the work.

Appendices

- Appendix A: Technical implementations: This appendix contains diagrams and more technical information about the app, server and classification system developed for this work.
- Appendix B: Feature plots: Plots of feature distributions.
- Appendix C: Instructions: The instruction sheet given to participants using the app.

Chapter 2

Background

The relevant background information is presented in this chapter. First, emotion theory is briefly outlined together with the definition of emotions that will be used in this thesis. Then follows information relevant for data collection and classification. Finally, an overview of other related studies is presented, and it is shown how this work adds to what already exists.

2.1 Emotions

As a starting point we need a definition of *emotions*. In daily speech equivalent to feelings, they are within the field of psychology considered to have three components [37]:

- Specific feelings associated with the emotion
- Physiological changes in the person experiencing the emotion
- Inclinations to act in a specific way

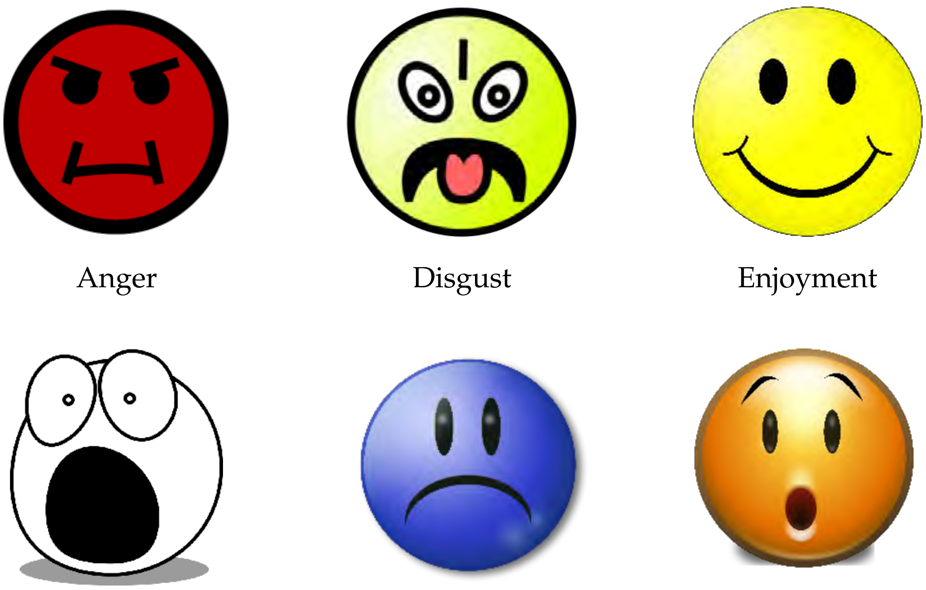
This understanding of emotions obviously also includes changes in the movement patterns, which is what we are concerned with in this work. A key difference between emotions and the related concept of *mood* lies in the duration. While mood is considered long-term, emotions are typically more intense and have a duration that can be limited to only a few seconds or minutes [20, 73].

Another way to view the difference between the two concepts is to see mood as a weaker and more persistent state of affect, which generally is not clearly visible to others. Mood is typically generalized into positive and negative mood. Emotions, on the other hand, can be considered more specific, spontaneous and short-term, and they will to a greater extent be visible on the person experiencing the emotions. Examples of emotions can be *happy*, *angry* and *surprised*. The terms are closely related, however, and mood tends to affect which emotions a person will experience (and vice versa). See table 2.1 for an overview on the two concepts.

Emotions are classified either as discrete categories of basic emotions or on axes in several dimensions [37].

Table 2.1: Overview of mood vs emotions

	Mood	Emotions
Duration	Long-term (hours/days)	Short-term (seconds/minutes)
Expressiveness	Lower	Higher
Intensity	Lower	Higher
Cause	Non-specific	Something specific
Examples	Usually generalized into positive mood and negative mood	Enjoyment, surprise, anger, fear etc.



Anger Disgust Enjoyment

Fear Sadness Surprise

Figure 2.1: Results of Google image searches on Ekman's six basic emotions [24]

2.1.1 Discrete categories of emotions

Corresponding to the common view of emotions, the discrete classification has been frequently discussed, e.g. by Ekman [19]. He defines the basic emotions *anger*, *disgust*, *enjoyment*, *fear*, *sadness* and *surprise*[1]. These emotions are very common, and most people have some kind of understanding of what they mean and how they are expressed. Figure 2.1 shows some of the first results when searching with Google for each of Ekman's six basic emotions in turn. Although the emoticons are very basic, they are able to show some of the facial expressions commonly associated with the different emotions.

Another list of emotions was compiled by Tomkins [70], defining the

[1] Ekman later adds to this, resulting in a list of 15 emotions[20]

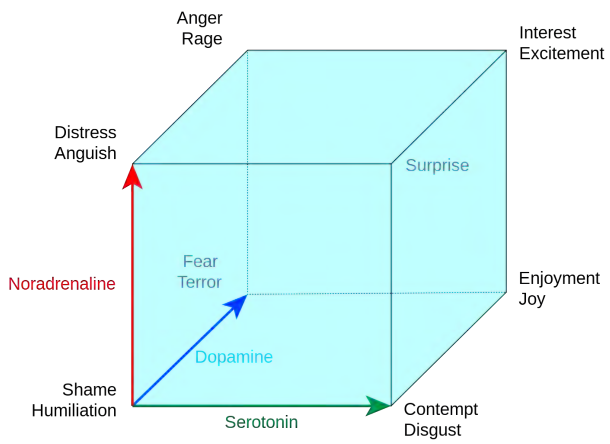


Figure 2.2: The Lövheim model of the connection between neurotransmitters and emotions

nine basic emotions of *interest*, *joy*, *surprise*, *distress*, *fear*, *shame*, *contempt*, *anger* and *disgust*. Tomkins' list is the basis of more recent research into the connection between emotions and neurotransmitters by Lövheim [39], see figure 2.2, a model that is recently used in affective computing research [26, 35]. While these selections seem rather limited, they encompass a fairly high portion of the emotional states a person can experience.

2.1.2 Emotions in two dimensions – A Circumplex model of affect

A dimensionality-based model of emotions was suggested by Russel [59]. Named *A Circumplex model of affect*, it seeks to model the different kinds of emotions on a two-dimensional chart. Figure 2.3 shows a simple drawing of the main idea, using the two dimensions *pleasantness* and *arousal*. These two dimensions have been shown to carry the majority of the information related to the perceived difference between various emotions [60].

Each emotion will be scored on the two dimensions, e.g. *angry* might get a high score on arousal and a moderately low score on pleasantness. This way it is possible to extract common terms from the dimensional model.

Another model of emotions is based on the dimensions of positive affect and negative affect. Positive affect can be expressed in generally positive terms like enthusiasm, alterness, energy and concentration, whereas negative affect is characterized by the extent of which a person feels anger, fear, contempt, nervousness and other generally negative emotions [74]. While the two terms immediately sound like opposites and thus would appear to belong on the same dimension, they are repeatedly shown to be the most important dimensions in studies of self-reported mood [75].

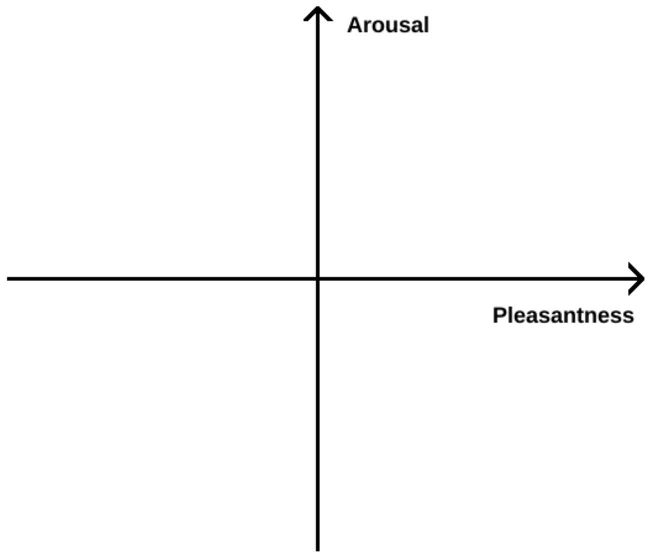


Figure 2.3: A Circumplex model of affect

2.2 Smartphone-embedded sensors

Modern smartphones have a high number of sensors available, in addition to software-based recordings of how the phone is used. For Android phones [3], the following sensors might be relevant for the task of identifying emotions:

- Motion detection (accelerometer)
- Rotation detection (gyroscope)
- Position detection (accelerometer and magnetometer)

Both the accelerometer and the gyroscope provide measurements for each axis in a 3-dimensional coordinate system, see Figure 2.4, and for this project we will focus on accelerometer readings. When information from the accelerometer is requested, the output is a vector consisting of the acceleration in each of the three directions x, y and z:

$$\vec{A} = \{a_x, a_y, a_z\}$$

This means that ideally, the accelerometer embedded in a phone lying completely still on a flat surface should output the values $A = \{0, 0, \pm g\}$. However, accelerometer data is not completely accurate, so this is not the case. The noise can easily be observed by leaving the phone completely still while looking at the output from the accelerometer. The values will not be constant, but rather change between each reading. This inherent inaccuracy is a challenge when it comes to the quality of the recorded data.

Accelerometers are generally not used for detecting emotions, or affective data in general, but rather for measuring movement and orientation. While affective data cannot be directly accessed through the accelerometer,

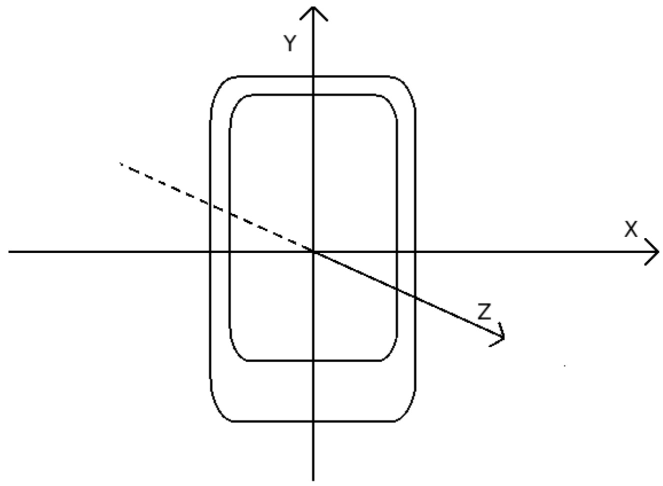


Figure 2.4: Coordinate system of sensors in a smartphone

it explicitly outputs the forces that have been exerted on the device. This allows us to decide the orientation of the device and how it has been moving. Later in this chapter there will be examples of how this in turn can be used to extract information about a person's emotional state.

A Google Nexus 5 was used for initial testing. It has a combined accelerometer and gyroscope [45]. With a potential sampling rate of up to 4000Hz [46] it is reasonable to assume that the sensors can provide a sufficiently fine-grained set of data. In practice however the sampling rate does not reach such high frequencies, but it should not be necessary anyway. The people who were recruited to participate in the data collection for this work had many different smartphones, and they all managed to output the decided sampling frequency. A discussion of sampling frequency is found in the following section.

2.3 Sampling frequency

When recording sensor data from the phone, it is necessary to set a sampling frequency. The choice of sampling frequency might impact the accuracy of classification, and the most useful rate has to be determined through trials. The initial sampling frequency used for testing was arbitrarily set to approximately 15Hz [2].

However, as the sampling frequency obviously impacts the quality of the data, it might be interesting to have as high sampling frequency as possible. One disadvantage of this approach is power consumption; more frequent sampling requires more power. The processing of data also takes more time, so we don't want to record unnecessarily fine-grained data.

Bonomi [7] reviewed some of the sampling rates used for the purpose of identifying activity. He notes that good results have been achieved in this context with sampling rates in the range between 20 Hz and 50 Hz. It

[2] Sensor.SENSOR_DELAY_UI[2]

is also shown that 10 Hz is sufficient to capture 95 % of the acceleration of the body center for someone who is walking.

Bonomi conducted his own experiment where the participants were equipped with an accelerometer placed on the lower back. They were then instructed to perform a series of movements and states without movement. Decision trees were used for classification, and different sampling rates between 1.25 Hz and 20 Hz were compared. This showed little difference between sampling rates of 5 Hz, 10 Hz and 20 Hz, but the results dropped significantly for even lower rates.

This makes sense when taking into consideration the Nyquist-Shannon sampling theorem: "If a function contains no frequencies higher than W [Hz], it is completely determined by giving its ordinates at a series of points spaced 1/2W seconds apart." [64] Because of the repetitive nature of the various movements used in the study, a sampling rate of 5 Hz is likely to capture a lot of the relevant information. For e.g. normal walking with an average speed of approximately 1.5 m/s, a walking cadence of 1 step/second is normal [61]. A step is considered the movement between the time a foot hits the ground and the time the same foot hits the ground again. With each foot being moved once per second, 5 Hz should be just within the limits of the Nyquist-Shannon theorem when it comes to the frequency of steps.

While such a low sampling frequency works well for capturing certain activities and gains the advantage of reduced power consumption and faster classification, it might not be the best option when classifying emotions. We are not only interested in the major movements indicative of which activity the user is performing, but also the minor body movements that potentially can tell us something about the emotional state of the user.

Therefore, it was decided to change the sampling rate to approximately 200 Hz [3] in order to get as accurate data as possible. This can also matter e.g. when it comes to step segmentation. With a high sampling frequency it seems easier to segment an accelerometer recording into individual steps, which is relevant for the feature extraction.

However, using such a high sampling frequency created some challenges. While the Nexus 5 model generally managed to output data for this frequency, sometimes a significantly lower frequency was used when recording data, without the introduction of any changes to the code. It was also apparent that the output from other smartphone models was not necessarily 200 Hz when run with the chosen parameter. Therefore, the sampling rate was eventually chosen to be 50 Hz, which appears stable across all encountered recordings and smartphones.

The higher power consumption that comes with high sampling frequency is not a problem in this work, as that frequency is only used when the phone is actually recording data. Otherwise a sampling frequency of 1 Hz is used for movement detection. Other potential disadvantages are the increased requirements for storage and network transfer, but the amount

[3] Sensor.SENSOR_DELAY_FASTEST[2]. The exact value is determined by the specific accelerometer and phone, and the Nexus 5 outputs 200 Hz.

of data is very small even with the highest sampling frequency.

2.4 Classification

Generally speaking, when we *classify* something, we group it alongside similar objects or concepts according to certain criteria. In this context, however, while still carrying the same meaning, classification can be described as a more specific process. Starting with the data collection, and ending with the classification itself being performed, the following steps are involved:

1. Data collection.
2. Preprocessing of data. Raw data might contain noise, or it has to be represented in other formats than its initial form.
3. Feature extraction.
4. Feature selection[4].
5. Classification. The final step takes the chosen features along with class labels as input, and a *machine learning* algorithm trains a classifier to give the correct labels for as much of the training data as possible. The result can then hopefully be generalized to new data.

The details of how these steps are performed in this thesis can be found in chapter 3 (step 1-3) and chapter 4 (step 4-5).

Feature extraction

Feature extraction is the process of transforming data into a set of features. A *feature* is a property of the data, either the raw data itself or calculated through the use of mathematics and statistics. That means that there is no limit to the number of possible features that can be extracted from a data set, but characteristics of the data can suggest what kind of features we are interested in. An example of a relevant feature extracted from accelerometer data is the average acceleration, calculated per axis or combined for all axes. The specific features used in this work are chosen mainly based on what is used in other relevant studies, see section 3.4 for details.

Feature selection

While the set of possible features is theoretically infinite, only a subset of the features is likely to provide good discriminatory properties. The process of feature selection attempts to identify the good features and discard the bad ones. Successfully choosing only the best features can improve the classification results while simultaneously reducing the time

[4] This can also be a part of the feature extraction process itself, but in this work it is considered a separate process.

spent training the classifier. Improvement is especially likely if the data set is limited and there are many features, due to the risk of *overfitting*, adapting the algorithm too much to the training set so that the ability to generalize is reduced.

Therefore, some kind of feature selection should be performed. Particularly because the data set collected for this project consists of recordings collected from a very low number of people (see section 4.1 for details), the risk of overfitting and thereby reduced generalization is high.

One way to perform feature selection is through a method called *recursive feature elimination (RFE)*. As the name implies, the idea is to start with all features and recursively remove the feature that contributes the least (i.e. has the lowest absolute weight) to the classification. This is repeated till the desired number of features remain. For this project, rather than deciding to use a specific number of features, the following criterion will be used: Remove features as long as there is no drop in classification accuracy. This means that the training set will be classified with the lowest number of features that still classify with the highest possible accuracy. RFE is accessed through the Scikit-Learn Python framework [56], and the implementation details can be found in section A.3.1.

Confusion matrix

Classification is often evaluated through the use of a *confusion matrix*. This is a way to see the distribution of correctly and wrongly classified samples, and for this project it takes the following form:

$$\left[\begin{array}{c|ccc} & \mathit{Low} & \mathit{Neutral} & \mathit{High} \\ \hline \mathit{Low} & \mathbf{LL} & \mathit{LN} & \mathit{LH} \\ \mathit{Neutral} & \mathit{NL} & \mathbf{NN} & \mathit{NH} \\ \mathit{High} & \mathit{HL} & \mathit{HN} & \mathbf{HH} \end{array}\right]$$

The columns of the confusion matrix denote correct classes, while the rows denote the classification output. Most interesting is the diagonal, where the number (or percentage) of correctly classified samples are listed. This means that e.g. the position marked *NN* is the number of samples that actually belong to the neutral class and also are classified as neutral. *HN*, on the other hand, are actual neutral samples that are classified as high, while *NH* are actual high samples classified as neutral. This way we can both find the classification accuracy and see which classes are the most challenging to classify correctly.

2.5 Machine learning

The idea of machines being able to undergo a learning process has existed for a long time. The computer scientist Alan Turing discussed this concept in his paper *Computing Machinery and Intelligence* [71], published in 1950:

> *Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this*

were then subjected to an appropriate course of education one would obtain the adult brain.

Children (and adults, for that matter) learn from experience. When they enter an unknown situation, they need to decide upon a course of action. The choice of action is based on a combination of previous experience and theoretical knowledge, and the result of the action they commit to will help them make decisions in similar circumstances later.

Machine learning tries to copy some of the mechanisms used in biological learning in order to allow the computer to make better decisions. This consists of making the computer, based on some kind of feedback to its decisions, adapt its behaviour to give output closer to what is considered the correct output [42]. An example can be a computer playing a game against a person. Having an initially poor performance, the computer might eventually learn better strategies for playing the game and end up defeating the human player. The computer can then bring the playing strategies it has developed into a game with another human or computer opponent, taking advantage of the learning from previous games in order to perform better later.

2.5.1 Categories of machine learning

Marsland [42] describes four types of machine learning:

- Supervised learning: Given a set of inputs labelled with the known class, the algorithm tries to use information it knows is correct to generalize to other cases.
- Unsupervised learning: The algorithm attempts to identify patterns in the input data. In this case no correct answers are provided, so unsupervised learning algorithms search for similarities and group the different inputs accordingly.
- Reinforcement learning: In this case the correct answer is known, but the algorithm isn't told specifically what it is. It is only informed that it is right or wrong and has to continue searching for the correct answer.
- Evolutionary learning: Evolutionary learning algorithms are inspired by biological evolution. They use a population of solutions, where a fitness metric is used to evaluate the quality of the suggested solution.

For this work supervised learning is the natural choice, due to the fact that it is meaningful to use predefined classes for the classification. The remainder of this section will briefly describe three different classifiers whose performance will be compared, all of which are common choices in classification used on accelerometer data or related to affective computing: Decision tree[5], support vector machine[6] (SVM) and multilayer perceptron

[5] Used by e.g. [6, 7, 16, 25, 33, 77]

[6] Used by e.g. [5, 16, 23, 25, 33, 76, 77]

(neural network, MLP)[7]. Evaluations of performance on activity recognition, which also uses accelerometer data and is related to emotion recognition, indicates that these three classification methods are good choices [53].

In theory, we could try to classify the emotional state of the user through unsupervised learning. Still using the app *Emotions* (see section 3.2.3) for data collection, input data could be collected at appropriate moments without asking the user to report his/her emotional state at the time of recording. The classification is then performed with a *clustering algorithm*, an unsupervised learning algorithm where the data is grouped into clusters based only on the properties on the data itself and without the use of any predetermined classes. This might yield a reasonable division of the data, but we won't be able to determine specifically which kind of emotional state is related to each of the resulting classes.

With supervised learning, on the other hand, we need to ask the user to report his/her emotions, in order to collect a training set that can be used to train a supervised classifier. Because we want to know the emotional state the user is experiencing, and not just differentiate between different, but unknown states, this immediately sounds like a better choice. A possible downside to this approach is that the input data might be more vulnerable to inaccuracies in the self-reported emotions.

2.5.2 Decision tree

Decision trees are considered one of the more easily understood types of machine learning. Classification also tends to be performed very quickly once the classifier is trained, as the complexity of a balanced binary tree is $\mathcal{O}(logN)$, where N is the number of nodes in the tree. Although the tree might not necessarily end up balanced, the lookup will still be performed quickly. Additionally, it is simple to illustrate how the classification decisions are made.

Decision trees are constructed starting with a root node. This node represents the *most informative* feature [42]. That is, the feature that has the highest entropy. A feature that can split the samples into two parts, where each comprises 50% of the samples, is more interesting to choose than a feature that gives a 90%/10% split. This is due to the higher entropy gain involved in the first case, and it also contributes to creating a more balanced tree.

Once the root node is decided, that feature is removed from the set of features, and the total entropy gain from each feature is calculated. Again the feature providing the most information is chosen, and, using an early algorithm called *ID3* [54], this is repeated till all features are used or there is only one remaining class. This method risks overfitting, as even features with minimal value are used. Such features might have a negative impact on data outside of the training set, as the (minor) discriminatory ability found could simply be due to noise in the limited data set used for training. Newer algorithms use some kind of *early stopping*, validating the

[7] Used by e.g. [16, 25, 33]

tree with a validation set after each iteration and halting when no further improvements are made, or *pruning*, using a similar principle on a tree after it has been built.

Figure 2.5 illustrates how a decision tree can be used for classification, starting at the top node and following the tree down till a leaf node is reached.

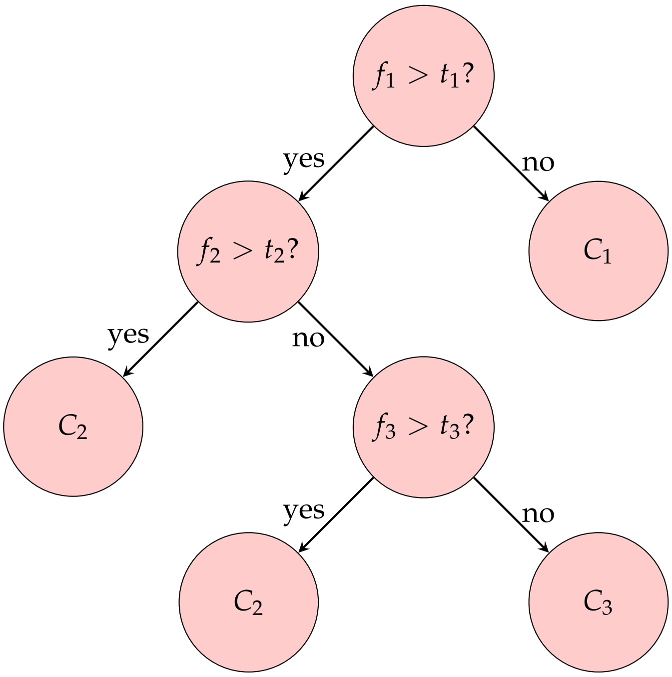


Figure 2.5: A simple decision tree with the continuous features f_1, f_2, f_3, split at thresholds t_1, t_2, t_3 and classes C_1, C_2, C_3. Here f_1 is calculated to be the most informative feature and is thus placed at the root of the tree.

2.5.3 Support vector machine

A non-linear classification problem cannot be solved by a linear classifier. However, it is possible to transform the non-linear data set in a way that allows it to be classified linearly. One approach to this problem is to use *kernel methods* [42], a way to determine which dimensions to use for the transformation. Kernels can be different kind of functions, e.g. polynomials. As an example, consider a kernel function consisting of all polynomials up to the 4. degree. For a single feature f, the transformation would be as follows:

$$\Phi(f) = \{1, f, f^2, f^3, f^4\}$$

With such a transformation applied, it might be possible to separate the classes linearly, as shown in the example in Figure 2.6. And this is the basis for SVMs, together with the observation that a line separating two classes[8] should be placed in a way that leaves as much space as possible between the line and the closest data points. This is illustrated in Figure 2.7. These data points are called support vectors and are the most important ones for training the classifier.

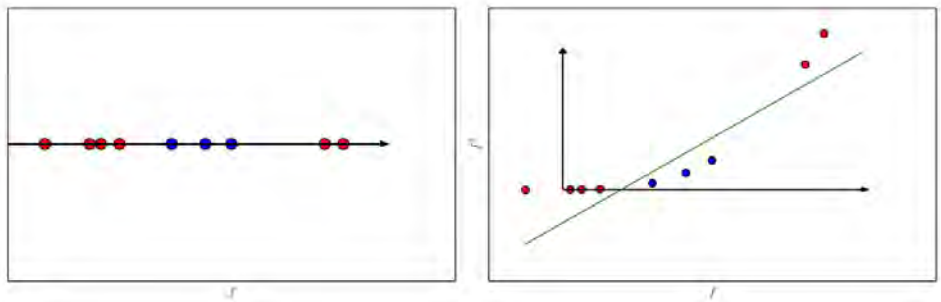


Figure 2.6: To the left is a plot of the feature values f, and we see that it is not possible to draw a linear decision boundary between the two classes. To the right, such a boundary can be drawn, because the feature f^3 has been added.

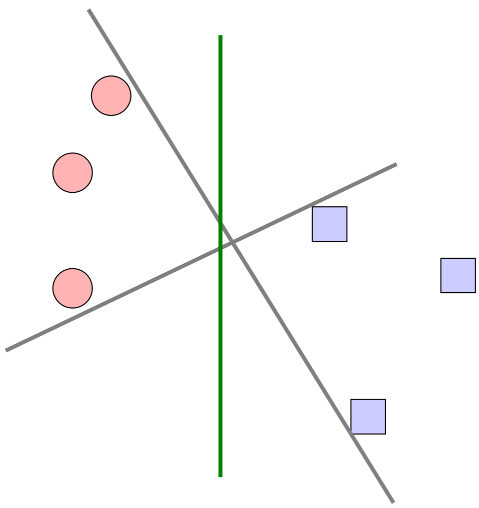


Figure 2.7: All three lines separate the two classes, but the vertical one is best. The other lines are located just next to the data points, giving them no margin to each side, while the vertical line has a large margin.

[8]SVMs are binary, so in order to solve multi-class problems, multiple binary SVMs have to be trained.

2.5.4 Multilayer perceptron

The multilayer perceptron (MLP) is based on the functions of the brain. Comprised of a large amount of *neurons*, these neurons are connected through *synapses*, connections between two neurons. The neurons have one assignment: decide whether or not to fire. This decision is based on their input, which typically is the output from other neurons. What allows learning to happen, and the essential part of why neurons can inspire machine learning, is the fact that the connections between the neurons can change based on experience.

When it comes to the MLP, the connections are implemented as a graph consisting of multiple parts [42], as shown in Figure 2.8:

- A set of input nodes, the number of which is equal to the number of features.
- Weights determining the connection (synapses) between nodes (neurons).
- One or more hidden layers of nodes.
- A set of output nodes, representing the classes involved.
- Each layer of nodes, apart from the input nodes, also have a set of *bias* nodes, which have weights similar to other nodes, but their output is a constant value.

For the training phase we have a set of feature vectors generated from samples with known classes. These vectors are fed to the network through the input nodes. Then, using the weights belonging to each synapse, the activation of the neurons connected to the input nodes is calculated. Then those new values are sent to the next layer of neurons, and new values are calculated. This is repeated till the output nodes are reached.

If the output has been given correct labels, i.e. the same as the labels belonging to the feature vector that was initially used, the training is finished. Otherwise, which is usually the case for many iterations, the weights have to be adjusted. In order to do that, the difference between the desired value and the actual value is calculated. This error is sent back to the network through a process called *back-propagation*, where the weights are adjusted iteratively in the opposite order of which they were encountered on the way from input nodes to output nodes. This has to be done in order to decide which weights should be changed. For an MLP with one hidden layer, the following would happen:

- Calculate error at the output nodes.
- Calculate error at the hidden layer nodes.
- Update weights pointing towards the output nodes.
- Update weights pointing towards the hidden layer nodes.

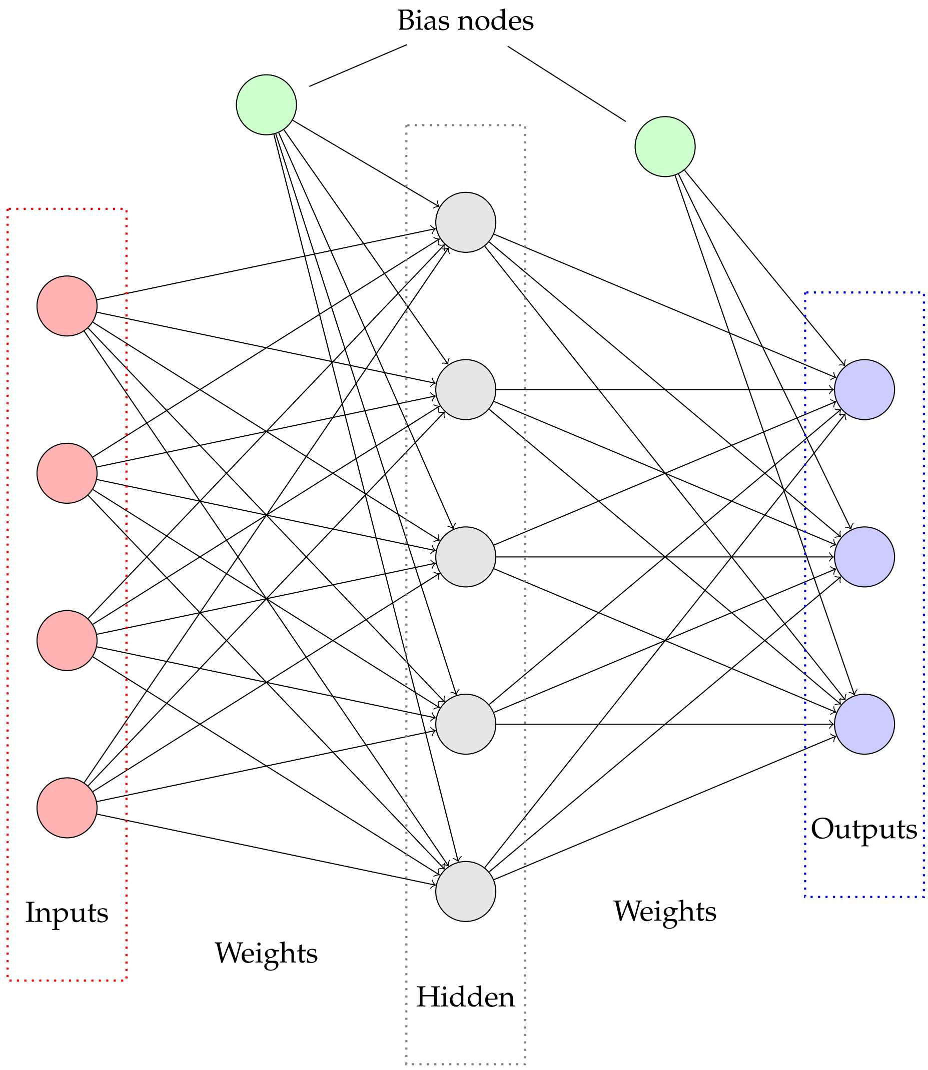


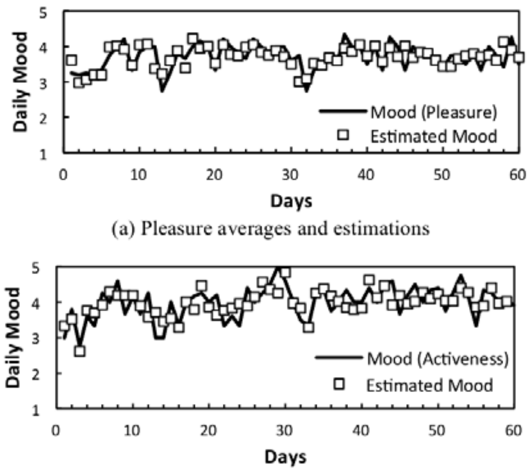
Figure 2.8: Multilayer perceptron

This way of going forwards and then backwards in the network is repeated multiple times for all inputs. In order to decide when to stop this process, a validation set is used, and training is stopped once the error of the validation set has reached its minimum and starts increasing again. This is a sign of overfitting and a strong indication that the training should stop.

2.6 Previous work related to detecting affect

Research has been done on mood recognition as well as emotion recognition. These two terms, while frequently used interchangeably in the literature on affective computing, have different meanings, as explained in section 2.1 and also discussed in affective computing research [38, 78].

An overview of some relevant work can be seen in table 2.2. Most of the reviewed studies involve affect detection. In some of them an



(a) Pleasure averages and estimations

(b) Activeness averages and estimations

Image credit: *Likamwa et al. [38]*

Figure 2.9: Emotions predicted by Likamwa et al.

accelerometer or other sensors are involved, but many different sources of input are used. A study concerned with activity detection is also included, as accelerometer-based activity detection has shown high accuracy and has similar challenges when it comes to preprocessing of data.

2.6.1 Smart phone based research

MoodScope – mood prediction based on phone usage patterns

LiKamWa et al. designed a system called MoodScope [38], inferring mood from the way a person uses his phone. The system is based on the idea of usage patterns being different when the user is in a different mood. The data analyzed are logs of text messaging, calls, email, browsing, app usage and location.

In order to collect data for the study, they created an application which prompted the user for his/her mood at least four times each day. Based on the circumplex model (see section 2.1.2), the user had to set a value on two scales, each divided into five discrete options, indicating his/her experienced mood. This data was put together with the phone usage patterns, and methods of machine learning were used to find a way to infer the user's mood.

The results from this study are promising, yielding 66% correct classification immediately, and 93% after two months of additional training on the specific user. Figure 2.9 shows the high accuracy. What is studied here, however, is the long-term mood, and a similar approach is probably not suited to infer the user's short-term emotions, as the usage logs only help identify the daily mood average. The long period of training required for getting good results is also a significant disadvantage when it comes to practical use.

Table 2.2: Work related to emotion detection

Summary	Sensors	Conditions	Categories	Accuracy
MoodScope – mood prediction based on phone usage patterns (LiKamWa et al.)	None, only phone usage data	Personal device, long term	Circumplex model	66%–93%
Emotion detection taking into account user's sitting posture (Hossain et al.)	Smartphone accelerometer	Controlled environment	Neutral, stressed, excited	51%–70%
MoodMiner – mood prediction based on sensor data and smartphone usage (Ma et al.)	Smartphone gps, accelerometer and usage data	Personal device, long term	Three dimensions: displeasure, tiredness, tensity	Below 50%
Activity monitoring based on accelerometer data (Zhang et al.)	Accelerometer in smartphone strapped to belt	Controlled environment + at the participant's workplace	Six states descriptive of activity or no activity	63%–83%
Emotion detection based on bracelet-embedded accelerometer (Zhang et al.)	Accelerometer worn on wrist and ankle.	Controlled environment	Angry, neutral, happy	81%
Accelerometer- and pressure sensor-based emotion detection related to sitting posture (Shibata and Kijima)	Accelerometers attached to multiple parts of body + pressure sensors on chair	Controlled environment	Three dimensions: pleasantness, arousal, dominance	Promising for the first two dimensions
Emotion prediction based on movements (Bernhard et al.)	Data from motion capture (suit + cameras)	Controlled environment	Neutral, happy, angry, sad	50%–81%

Image credit: *Hossain et al. [6]*

Figure 2.10: The eight sitting positions used by Hossain et al.

Tang et al. [68] did something similar as part of a study on social networks, taking into consideration the users' location, phone calls and text messages. Tang et al. split emotions into five discrete states: *wonderful, good, neutral, bad* and *terrible*. In their study the users had the freedom to choose when to input their emotional states; either immediately as their mood changes, or later. The paper acknowledges possible problems of users providing inaccurate labels on their emotional states.

Emotion detection taking into account user's sitting posture

The accelerometer can be used to track the minor movements that occur while the user is using the phone. Hossain et al. [6] take advantage of this and try to predict the emotional states *neutral*, *stressed* and *excited* based on these movements. As the accelerometer records data from the three dimensional axes separately, the recordings from each axis can differ greatly based on the position in which the user is sitting.

Hossain et al. designed an android application for the purpose of data collection. Their test users would install the app and run it for a specific amount of time. First the app would request two entries from the user: their sitting position (figure 2.10) and their current mental state: *neutral, stressed* or *excited*. For the duration of the data collection, the users were instructed to remain in the current sitting position. Their task was simply to spend some time doing normal phone activities, i.e. writing messages and browsing the internet. In the meantime, the app would record data from the accelerometer in the background, and the recordings together with the self-reported sitting position and mental state would make up the input data.

Emotional state	Count
Neutral	5282
Stressed	7638
Excited	4842

Figure 2.11: Distribution of emotional states

This data was then classified using decision trees[9]. The results were not impressive, but they got decent classification rates between 51% and 70% when trying to predict which of the three emotional states the user was in. It indicates that the accelerometer can be used to discover minor differences in movement during phone usage that are connected to the emotional state of the user.

When looking at the recorded data, it is interesting to observe that the test users report a somewhat surprising distribution of emotional states. Figure 2.11 lists the distribution based on 17762 lines of test data. This shows a very high number of users reporting to be stressed, and other researchers (e.g. Watson [73]) report a much higher frequency of positive emotions than negative in self-report data. Why the opposite is the case here might have something to do with the choice of test users, as the students who participated in this study could potentially have been stressed about their exams, or there might be another explanation. It is difficult to tell whether or how it has impacted the data collected in the study.

MoodMiner – mood prediction based on sensor data and smartphone usage

Ma et al. [41] conducted a study similar to what Likamwa et al. [38] did. Named *MoodMiner*, the system tries to predict the daily mood measured in three dimensions – *displeasure*, *tiredness* and *tensity* – based on data from different sensors as well as communication logs. The following features are used:

- Location (from gps sensor)
- *Micromotion*: "a user picks up the phone and does nothing useful for no longer than a few seconds" (from accelerometer)
- Activity: Whether the user is walking, running, sitting or standing (from accelerometer)
- Communication frequency: The number of times the user has called or texted per day

MoodMiner records sensor data all the time when the app is active and stores the data on the phone. The users can decide themselves when to

[9] More specifically, the algorithms named J48, Random Tree and Simple Cart

report their mood, and that data is also stored. If the users report their mood multiple times on the same day, a weighted average is used (later reports have higher weights). Each day, the data is uploaded to a server, together with the recent communication logs.

As opposed to Likamwa et al., *MoodMiner* also uses the accelerometer. Due to the relatively low battery cost of the sensor, it is monitored continously, but the data is only stored when the difference between two consecutive readings is higher than a given threshold. Otherwise there is no significant motion, and it is not necessary to store the data. Additionally, the accelerometer is used to determine whether or not the gps sensor should be activated, as the gps sensor requires much more battery.

The results are compared to the self-reported data along the three dimensions in order to evaluate the classifier. In average the results are just below 50%, with the results for pleasantness a little above 50%. This is significantly worse than the results MoodScope [38] achieved, and it might have something to do with calculating fewer features, having less users in a shorter time frame and/or not developing a personalized model.

Activity monitoring based on accelerometer data

Zhang et al. [76] use accelerometer recordings to detect what kind of activity a person has been doing. They distinguish between the following six states:

- The four basic states of lying, sitting, standing and walking
- Gentle movement
- Transitioning between different postures

In order to collect data, the users had to wear a phone with an embedded accelerometer strapped to the belt on the side of their waist. That way the movements and orientation of the users could be recorded. The experiments were performed for a set amount of time in a laboratory where the users part of the time engaged in set activities, and also at the workplace. Video recordings and self-reports were used in order to validate the activity classification.

For the classification, Zhang et al. performed two steps. First, they classified periods of time as either motion or motionless. This was done through introducing a set of rules intended to identify various cases of e.g. noise while correctly deciding whether there had been movement during a period of time. As part of this set of rules they used a measure of change in acceleration, and a a similar measure will be used for data collection in this work (see section 3.2.3). The remaining part of classification was performed through the use of multiclass SVM classifiers, with the training set being formed by data from the first participant.

The average result of classification was 82.8%, using the described two-step classifier. In comparison, a single-step classifier where there was no initial classification of motion or no motion only achieved 63.8%, which shows the benefit of first determining whether there has been motion.

Emotion detection based on bracelet-embedded accelerometer

The output of an accelerometer attached to a person depends strongly on where on the body the accelerometer is located. Zhang et al. [77] performed a study where the accelerometer was worn on the wrist and the ankle. The results were generally better when using data from the one worn on the wrist.

In their study, accelerometer data was collected from 123 participants when they were walking in a controlled experiment. The experiment was divided into two parts; first a neutral state was compared to an angry state, and then a neutral state was compared to a happy state. In both experiments the participants started out walking for two minutes back and forth in a designated area. Then the participants would see a movie clip intended to induce an angry and a happy state of mind, respectively. Afterwards, they would resume walking for another minute.

During the walking, accelerometer data was recorded at a rate of 5 Hz. Due to the inherent noisy structure of accelerometer data, Zhang et al. then applied a moving average filter in order to reduce noise. Using 114 features, they applied principal component analysis and implemented different classifiers for the purpose of comparison.

The best result for classification between the three different categories of *neutral*, *happy* and *angry* was 81.2%.

Other studies based on smart phones

Rachuri et al. developed a system called EmotionSense [55] as a framework for social psychology research. Their main way of emotion prediction is through speech analysis, but they also involve other sensors like the accelerometer. Used simply as a binary state of *moving* vs *not moving*, the distribution of emotions is compared to the user's state of activity. They show that the users are much more likely to experience either of the emotions *happy*, *anger*, *sad* and *fear*[10] when not moving as they are when moving.

Adibuzzaman et al. used facial expressions and accelerometer-based measures of energy expenditure to detect emotions from mobile users [1]. The energy E was calculated from an accelerometer recording[11] in three axes as follows:

$$E = \int_{t_0}^{t_0+T} |a_x| + |a_y| + |a_z| dt$$

Some connection between the energy and the reported emotions was found, see figure 2.12, but it is not consistent with Russel's model (section 2.1.2), where e.g. anger should have a higher arousal state than happiness. This might also have something to do with the low number of participants (8) involved in the study.

[10] As opposed to *neutral*

[11] The raw data has been low-pass filtered in advance

Affective State	Energy(mean)
Anger	8.56E+00
Disgust	2.24E+01
Fear	4.12E+01
Happy	1.51E+01
Sad	4.28E+00
Surprise	3.56E+01

Image credit: *Adibuzzaman et al. [1]*

Figure 2.12: Average energy calculated from accelerometer values by Adibuzzaman et al.

2.6.2 Other studies

Accelerometer- and pressure sensor-based emotion detection based on sitting posture

Previously, emotion detection where the user's sitting posture was taken into account was reviewed (see 2.6.1). The posture itself can also give information about a person's emotional state, and this was examined by Shibata and Kijima [65].

In their study they used a chair equipped with various pressure sensors. Together with accelerometers attached to different places on the participants' bodies, they collected a total of 21 descriptors[12] of the participants' postures. This combined sensor input provided a lot of information about a person's sitting posture, and the study included 24 different postures (see figure 2.13).

In order to decide which emotions the various sitting postures expressed, they performed another experiment where the participants were sitting in the same postures, but without sensors. Instead, a group of observers scored the emotional expressions they observed the postures to contain on a list of adjectives. Having 16 adjectives (i.e. "excited", "calm", "happy" and other adjectives relevant for the expression of emotions) scored from 1 to 7, the final labeling was based on the observers' average scores.

The model of emotions used by Shibata and Kijima is based on three dimensions: *pleasantness*, *arousal* and *dominance*. While they concluded that the first two dimensions could to some extent be measured through the use of sensors, the results for the dimension of dominance were not good. This happened because the physical expression for dominance was not caught by the sensors used.

[12] 12 pressure sensors and three accelerometers attached to the participants' neck and lower arms, each of which had three axes

Image credit: *Shibata and Kijima [65]*

Figure 2.13: The 24 sitting postures used by Shibata and Kijima

Emotion prediction based on movements

Body language is frequently linked to a person's emotional state. Interpreting someone's body language is a very complex procedure, consisting of reading many small and large movements, of which some or most might be involuntary.

Instead of trying to read all signals at once, it is possible to break down compound movements into very basic movements. Bernhardt et al. [5] used such an approach when they tried to classify a person's emotional state as *neutral, happy, angry* or *sad* based on how they performed the motion equivalent to knocking on a door. This compound movement was broken down into four different parts:

- Raising the arm from hip to neck.
- Pushing the arm away from the body (knocking)
- Retracting the arm to the previous position
- Lowering the arm to the hip.

Having segmented the compound movement into these simpler movements, Bernhard et al. normalized them and extracted various features (maximum distance between hand and body, average hand speed, average hand acceleration and average hand jerk). Using machine learning they further attempted to classify the compound movement into one of the four aforementioned emotional states based on the features extracted from the simpler movements.

The data used for this study was taken from a prerecorded database of various movements performed by 30 different participants [40]. These movements were recorded with a motion capture system consisting of eight cameras and a suit worn by the participants. 35 markers were placed on the suit in order to give the cameras a set of reference points.

The results of this experiment were initially decent, with a recognition rate of 50%. However, when the individual movement bias – that is, the tendency of people to have certain individual characteristics influencing their movements in general – was taken into account, they achieved a recognition rate of 81%, which shows that even very simple movements can differ significantly depending on the emotional state of the person performing them.

A similar study was later conducted by Gong et al. [23]. Using the same source of data as Bernhardt et al, they implemented a new feature descriptor and improved the results not corrected for bias. The unbiased results did not improve, and they are still substantially better than the others, but also less practical due to the need for personalization.

2.6.3 Contribution of this project

The studies that have been described in this section generally show very promising results. Affective data has been collected by analyzing movement and various elements of smartphone usage. While these results are very interesting, there are some challenges when it comes to practical use. The following are some of the limitations amongst the mentioned studies:

- Analysis of phone usage data has to be done over a long period of time. While showing high accuracy on predicting a person's long-term mood, it cannot predict short-term emotions. For the purpose of mood monitoring it is effective when looking at periods of time of at least one day, but this approach fails to give good predictions for shorter time frames.
- Studies performed in controlled environments might cause the participants to move or otherwise act different than they do in their normal environments. Picard [49] discusses this problem and mentions e.g. the reluctance of some participants to express negative emotions in a laboratory setting.
- People generally do not want to wear extra gadgets. Requiring external sensors, e.g. accelerometers worn on specific parts of the body, or requiring that the phone is carried in a specific way, are not reasonable expectations in people's everyday life.

This project tries to approach the problem of emotion prediction in a more realistic way. It is reasonable to assume that most smartphone users carry their phone, usually in a pocket, when they move. If we can find a way to predict emotions solely based on how the person moves, as recorded by the smartphone-embedded accelerometer lying in a pocket, we have an approach that can be used by most smartphone users in a non-obtrusive way. This increases the likelihood of finding a practical use for the gathered information.

Chapter 3

Detecting emotions

Different ways of predicting emotions from smartphone data have been reviewed. In this work, the focus is on the movement sensors of the phone, more specifically the accelerometer. The following is an overview of the contents of this chapter:

- The model of emotions defined by the Circumplex Model of Affect (see 2.1.2) is the starting point for the data collection. Using that model, accelerometer data will be collected together with self-reported emotions from a set of participants. Details of the data collection follow in section 3.2.

- When considering how to collect data, I decided on developing an app that the participants can keep running in the background. The app will then request data when appropriate. Compared to the alternative of setting up a controlled test environment, this solution lets the participants use their own devices and allows us to record data from situations where they are not finding themselves in an artificial testing environment. This should give the study greater *ecological validity*, i.e. let the testing environment approximate the real world as much as possible, and hopefully provide data that is recorded as unobtrusively as can be expected. The specific functionality of the app is described in section 3.2.3.

- The raw data will be filtered and segmented into steps and periods of motion. features will be extracted from that processed data, and this is explained in sections 3.3 and 3.4.

- Finally, classifiers will be trained on the recorded data (see section 2.5.1 for selected types of classifiers). The experiments performed and results can be found in chapter 4.

3.1 Choice of model for emotions

The discrete classification of emotions described in section 2.1.1 gives a very clear distinction between the different categories of emotions. It also corresponds with how people generally understand the concept of emotions. However, it also potentially increases the number of classes we need to use for the purpose of classification[1], and it will be difficult to handle situations where the person experiences multiple emotions at the same time. Additionally, it might prove more challenging to gather test data from participants if the interface consists of a large number of alternatives. A simpler user interface will likely increase the probability of the participants making a report.

The discrete categories of emotions also generally do not include typical low-energy states, e.g. tiredness or serenity. Such emotional states are characterized by low activation and don't necessarily fit all the requirements for being labelled as emotions, but for the purpose of this project it is interesting to identify them as well rather than simply grouping them with the *neutral* state. At the same time, the selection of positive emotions is very limited in these models. For the purpose of letting the user find a suitable alternative, multiple other options might have to be included [73]. Finally, it has been shown that when only presenting a limited number of discrete emotions, there is a high probability of the user choosing to input a custom emotional state, if that is presented as a choice [11]. This indicates that such a discrete classification is insufficient to capture what the user wants to express. Therefore we will decide on the other approach and use a model that consists of two axes, thereby including all potential emotional states, for the cost of having to train the users in how to make a report in this "language".

Even though the model representing emotions in terms of positive and negative affect has been the most widely used dimensional model in studies of self-reported mood [74], it might not be the best choice for this work. Because we are using the dimensional model directly, and not as a measurement like PANAS[2], it might seem counterintuitive to the user to report his/her emotions on two seemingly contradictory dimensions. Therefore, the Circumplex model of affect will be used as a model for emotions in this work.

3.2 Data collection

3.2.1 Data collection while moving

The phone can be in many different states of movement and usage:

1. Lying still and not being used
2. Lying still and being used
3. Held in hand and being used while the user is stationary

[1] Depending on which set of emotions are chosen, as there are many differences in opinion on which emotions are considered the basic ones, e.g. [19, 27, 70].

[2] Positive and Negative Affect Schedule, a questionnaire designed to measure emotions on the dimensions of positive and negative affect by letting the participants rate the extent of which they have experienced each of a set of relevant emotions in the designated time period.

4. Held in hand and being used while the user is moving

5. Being carried and not being used while the user is moving

In order to increase the chance of making the classification work, this work will be limited to one of the mentioned situations. #1 and #2 are not relevant, as we are going to use the accelerometer for data collection, which obviously requires movement. #3 (and technically also #4) was initially tested[3], but due to apparent randomness in the data, we decided to postpone further testing and rather focus the project on #5. This is the basis of the functionality of the app *Emotions*, which is described in section 3.2.3. Later it would be interesting to revisit state #3 and see whether it is possible to get any results there, as previous research has shown some potential, see section 2.6.1.

3.2.2 Initial test for collection of sensor data

Initially – in order to more quickly get started – I decided to generate test data locally on my phone by just recording accelerometer data and manually labeling it with my emotional state at the time of recording. As the purpose of the project is to classify emotions based on accelerometer data when the phone is carried, it is necessary to record data from many different situations in order to find a way to classify different experienced emotional states.

Sensor data is accessible through the Android API, and for the purpose of data collection, the Android app that is developed for this work will use that API. For this initial testing, however, it was rather decided to use an app called *AndroSensor* [4].

AndroSensor provides a way to access data from all kinds of smartphone sensors and view them through the app interface, and after trying various similar apps it seemed like the most suitable one for this purpose. In order to record data, you have to click a button to start the recording, and then it keeps recording data from the selected sensors till the recording is stopped. Once a recording is finished, the output is sent by e-mail as a csv [13] file, giving easy access to the sensor data. This data can then be used for testing possible classification algorithms.

Initially trying this approach, it was soon abandoned because the recording had to be started manually in advance. This made it difficult to generate a realistic test situation, so it was deemed necessary to make an app for data collection first. That way, the person who is using the app does not have to think about any kind of data recording till the app suddenly requests input, and that should allow for higher quality of data. The specifics of the app are described in the next section, and an overview of the technical implementation can be found in Appendix A.

[3] The initial version of *Emotions* (3.2.3) supported this situation.

3.2.3 *Emotions* – App-based system for data collection

In order to get the required training data,the participants need a way to report their emotional state at the same time as the logs from the accelerometer are stored. For this purpose, an app which is based on the model of emotions outlined in section 3.1 has been developed. The purpose of the app is to gather accelerometer data and request the user to give his/her current emotional state, sending the data to a server, developed alongside the app, for storage.

An overview of the data collection can be seen in Figure 3.1, and more details can be found in Appendix A.

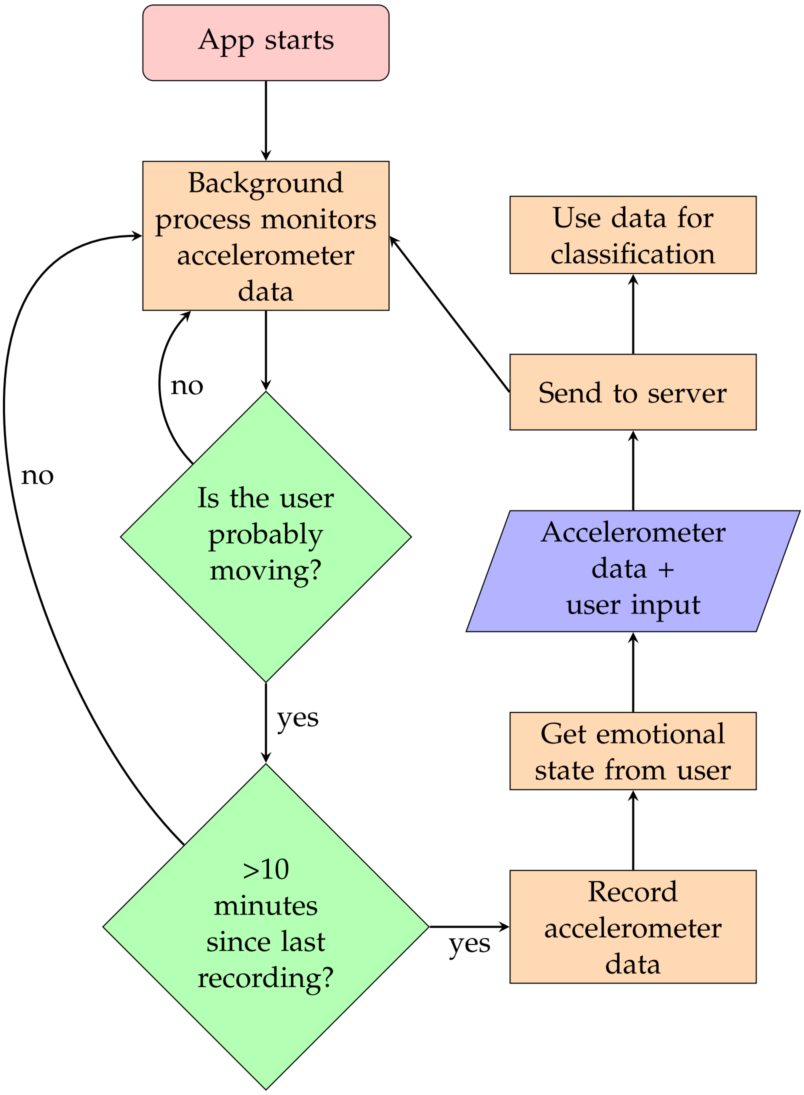


Figure 3.1: Overview of the data collection

App functionality before recording has started

Once the app is installed and opened, it launches a service that remains open till the app is closed by clicking the exit button. This service monitors the movements of the phone by fetching the value from the accelerometer once every second. The value is compared to the previous one, and if the difference is very small it is just counted as variation due to inaccuracies/noise in the accelerometer. A larger difference can indicate that the person is moving, and if a large difference is found in ten consecutive readings, i.e. over a period of ten seconds, it is probable that the phone is currently moving. This is one way to determine that we potentially want to make a recording, as we only want to do it when the user is walking. Once movement has been detected, the app determines whether or not to start recording according to the following criteria:

- If a recording has occurred recently (within the last ten minutes), nothing happens. Too frequent requests for information is a likely source of annoyance to the user, and that could lower the chance of getting data.
- Else the app intends to start recording and waits for two minutes. This delay is intended to handle situations where the user picks up the phone, puts it in a pocket and starts moving, or just puts it down again after performing some activity on the phone. If we start recording immediately, we will also include the initial movement where the user just picks up the phone. A delay of two minutes increases the chance of the user actually having started walking.
- When the two minutes have passed, the app checks whether the phone is currently being used, i.e. the screen being active.
- If it is being used, nothing happens, as that could indicate that the user just picked up the phone in order to use it.
- If the phone is not currently being used, the recording of accelerometer data starts.

App functionality during and after recording

Once the recording has lasted for a certain amount of time, i.e. 20 seconds, the recorded accelerometer data is tested in order to try to eliminate the recording if it has not happened in the correct circumstances, i.e. the user has not been moving all the time. In order to determine this, we look at the jerk (3.4) calculated from the acceleration. For entry i in a recording consisting of M entries, the jerk is

$$\vec{j}_i = \vec{a}_i'(t) \approx \frac{\vec{a}_{i+1} - \vec{a}_i}{\Delta t}$$
$$j_i = |\vec{j}_i|$$

When a person walks while carrying the phone in his pocket, the phone will be subject to plenty of movements where the jerk is relatively high.

This can be used to filter out the recordings where the user has not been moving all the time, as the jerk is generally lower when the person is using the phone and not moving. Therefore, we set a threshold j_θ and calculate the number of entries, denoted m, within a recording consisting of M entries, that exceed this threshold. If the relative number of such entries is higher than a certain amount p_θ, we assume that the person has been moving during the recording[4]:

$$m = \sum_{i=0}^{M} f(j_i) \text{ where}$$

$$f(j_i) = \begin{cases} 0 & \text{if } j_i < j_\theta \\ 1 & else \end{cases}$$

If $\frac{m}{M} < p_\theta$, the recording is considered invalid and discarded, without the participant getting any kind of notice. Otherwise the recording is valid, and the following happens:

- The recorded accelerometer data is stored while performing the next steps.
- The participant gets a notification and is asked to give his/her emotional state in a similar way to how it was done by LiKamWa et al. [38], placing a value on the two scales of the Circumplex model [59]. A screenshot of this interface, where the user has reported a neutral emotional state, is shown in Figure 3.2. Prior to receiving the app, the participants will get clear instructions on the meaning of the axes, as well as written instructions easily accessible within the app. Initially, sliders were used for this, see Figure 3.3, but it was changed in order to prevent possible response bias due to having preselected values. Switching to radio buttons ensures that the user has to make an active selection instead of potentially just leaving the values as they are. A review of relevant research into the use of sliders vs radio buttons performed by Roster et al. [58] indicates that sliders are perceived as more interesting, but also take more time to complete. They also mention possible biases when it comes to the starting point of the sliders, although there is little conclusive evidence that shows that one method is better than the other. For this purpose it does seem best to use radio buttons, as we want to reduce the chance of getting biased responses.
- The user input form has another check for the validity of the recording. If the participant answers *no* to the question of whether movement has taken place, the recording is discarded.
- If the participant confirms that movement has taken place, the user input together with the accelerometer data is sent to a server for further processing. Figure 3.4 shows an example of such a recording.

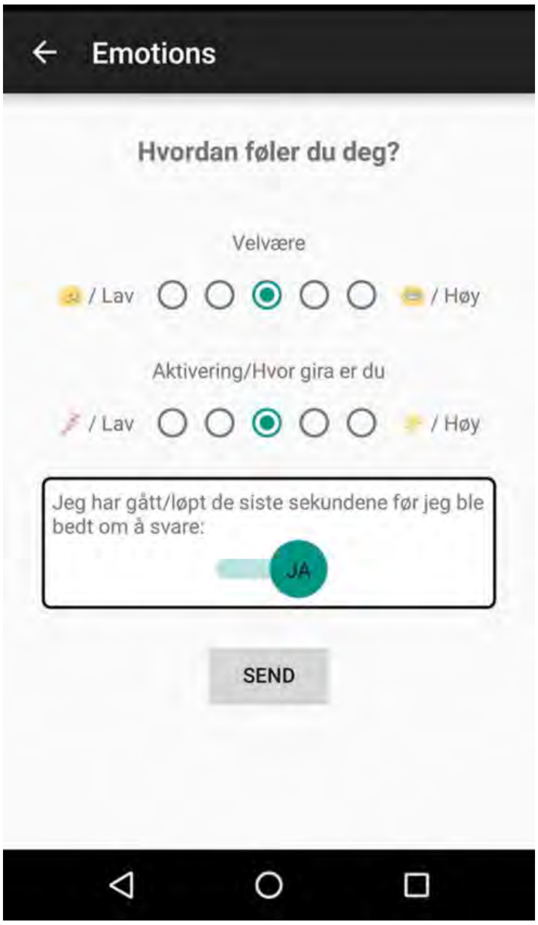


Figure 3.2: User input interface from the final version of the app

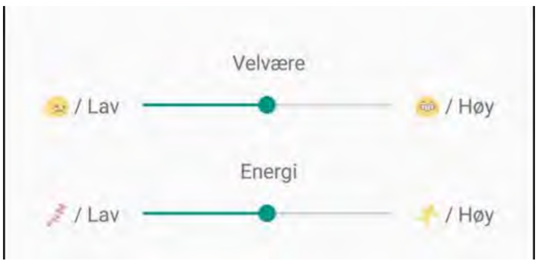


Figure 3.3: An earlier version of the app used sliders and a different translation of the arousal dimension

```
Id:5291431668149844285920884O7645
Monitoring:Movement
Intended time:5000
Time:5003
Size:1001
Pleasantness:2
Arousal:2
0.77420044, -2.554062, -11.891922
0.8533325, -2.8192444, -12.806641
0.75619507, -2.8994904, -13.139435
0.6590729, -2.9797363, -13.472229
0.25164795, -3.194809, -14.382675
```

Figure 3.4: A recording submitted by the app to the server

The recording in Figure 3.4 consists of the following:

- Id: A randomly generated ID intended to ensure anonymity of participants while also being able to keep track of the origin of the various inputs
- Monitoring: Indicates which of the usage states in 3.2.1 the app currently is recording
- Intended time: The app is set to record for a certain period of time (in milliseconds, ms)
- Actual time recorded (ms)
- Size: Number of lines of accelerometer data. Dividing size by time also gives us the sampling frequency (see section 2.3), which in this instance is approximately 200 Hz.
- Pleasantness: User-reported score on the *pleasantness* dimension of the Circumplex model (see section 2.1.2)
- Arousal: User-reported score on the *arousal* dimension of the Circumplex model (see section 2.1.2)
- Remaining lines: Entries from the accelerometer in each of the three directions, as illustrated in figure 2.4, where the first five out of 1001 lines are shown. With a sampling rate of 50 Hz, which is used in the final version of the app, and a recording time of 20 seconds, each recording should consist of approximately 1000 such lines.

Instructions given to the participants

The experiments in this study are not performed in a controlled environment. Because the participants are using the app over a longer period of time, potentially with days between each recording, it is important that they receive very clear instructions prior to using the app. After they have

[4]The same outcome can be achieved by e.g. shaking the phone, and for higher accuracy a better way to decide whether there actually has been movement should be implemented.

received instructions and installed the app, they are not expected to ask for instructions later, although they are informed that they can if they want to.

Prior to receiving the app the participants get the following information:

- An overview of the goal of the study, i.e. trying to detect emotional states based on movement recorded by the accelerometer in the smartphone.
- Clarification on the protection of personal data. Because no personally identifiable information is ever sent to the server, there are no privacy issues related to this study. The participants are informed that an ID is randomly generated in order to distinguish between different participants, but that ID is in no way connected to them or their device. They are informed that the data used for the study is accelerometer recordings together with what they input themselves.
- An explanation of the meanings of the two dimensions arousal and pleasantness. This seems to be very easily understandable by just looking at the user interface of the app, but as this model of emotions is unfamiliar to most people it seems prudent to explain this beforehand. This information is also given as instructions accessible from a button on the main screen of the app, see Figure 3.5.
- It is explained that only data recorded when the participant is walking is of interest, and that while the app tries to identify the correct state, it might not always succeed. Therefore they are also asked to state whether they have been moving whenever they input data.
- Finally, they are told that for the duration of participating in the data collection, they can leave the app open or close it by swiping it out to the side on the screen showing an overview of active applications. If they want it to quit properly, they need to click the exit button from the main screen.

3.2.4 Challenges

A number of challenges have been identified when working with the data collection. An overview can be seen in Figure 3.8.

Data quality

Two kinds of data are collected: self-reported emotions and sensor data from the accelerometer. Both sources provide some challenges when it comes to accuracy.

When the emotions of the participants are collected through self-report, we have to consider the accuracy of the given answers. People might for instance have a tendency to overestimate their emotions, both positive and negative, when giving a report some time after they have experienced the

Figure 3.5: Instructions accessible from the app

emotions [69]. Even when reporting emotions currently experienced, there may be errors, e.g. due to personal biases like *social desirability*[5] [9], but it doesn't necessarily make a huge impact on the results [18]. However, the literature is not conclusive in regards to whether we can rely on such self-reports of emotion [57].

Other potential problems related to the accuracy of the accelerometer data have also been identified:

- Noise causes the readings to fluctuate even when the phone is lying still. Filtering attempts to reduce the effects of noise.
- When the acceleration is relatively high, there seems to be some kind of maximum value output by the accelerometer. If this threshold is reached in one or more dimensions, we see in the output that this value keeps getting repeated rather than the actual value being shown (see figure 3.6). This threshold is higher than the acceleration normally experienced during the intended use (walking), but it should be kept in mind for the analysis/training.

[5]The inclination of a participant to answer according to what he thinks society or people present at the time of response might consider the desirable answer.

- A similar error can occur in normal use, which potentially is more serious. Possibly due to the way the phone handles an inactive state, the accelerometer on some devices provided less frequent updates than the actual frequency of the readings. This resulted in the same lines of data being repeated multiple times rather than each line of the recording being unique (see figure 3.7). A possible solution to this problem was to implement a *WakeLock*[6] [72], ensuring continued cpu activity throughout the recording. Additionally, reducing the sampling rate from the initial rate of 200Hz to 50Hz should help alleviate this problem.

```
19.368118, 0.65371704, -10.053986
19.368118, 0.6858978, -9.049942
19.368118, 0.42300415, -7.386978
19.368118, -0.037399292, -5.0321198
19.368118, -0.78474426, -1.9688721
19.368118, -1.6482391, 1.7616882
```

Figure 3.6: Accelerometer issue #1

Privacy

When collecting and analyzing personal data, in this case self-reported emotional data and accelerometer readings, the issue of privacy always has to be raised. In this work, the data is not only processed on the participant's phone, but also sent to a server where the data is later going to be used for training a classifier. Such transfer of data requires the privacy of the participants to be handled in a satisfactory way.

In this work, the privacy is ensured through complete anonymity of the participants. Each participant is identified only through a randomly generated id which is used to keep track of the origin of the various recordings. Because the data collection is implemented through a self-developed app and server, it is possible to control that no other potentially identifying information is collected. Therefore, there should be no privacy challenges in this context.

However, should the accelerometer-based detection of emotions be used in combination with other methods, further considerations would

[6] using the parameter PARTIAL_WAKE_LOCK [52]

```
7.5309615, -3.8977604, 2.834735
7.5309615, -3.8977604, 2.834735
7.5309615, -3.8977604, 2.834735
5.1367598, -6.4739213, 4.8985367
5.1367598, -6.4739213, 4.8985367
3.7576995, -8.576031, 6.009446
3.7576995, -8.576031, 6.009446
```

Figure 3.7: Accelerometer issue #2

have to be made. Other sources used for emotion detection sometimes contain personally identifiable information within the data itself, e.g. speech, text messages and location data. In such cases, it is important to take further steps to protect the collected data, and then it would also be necessary to obtain approval from the participants.

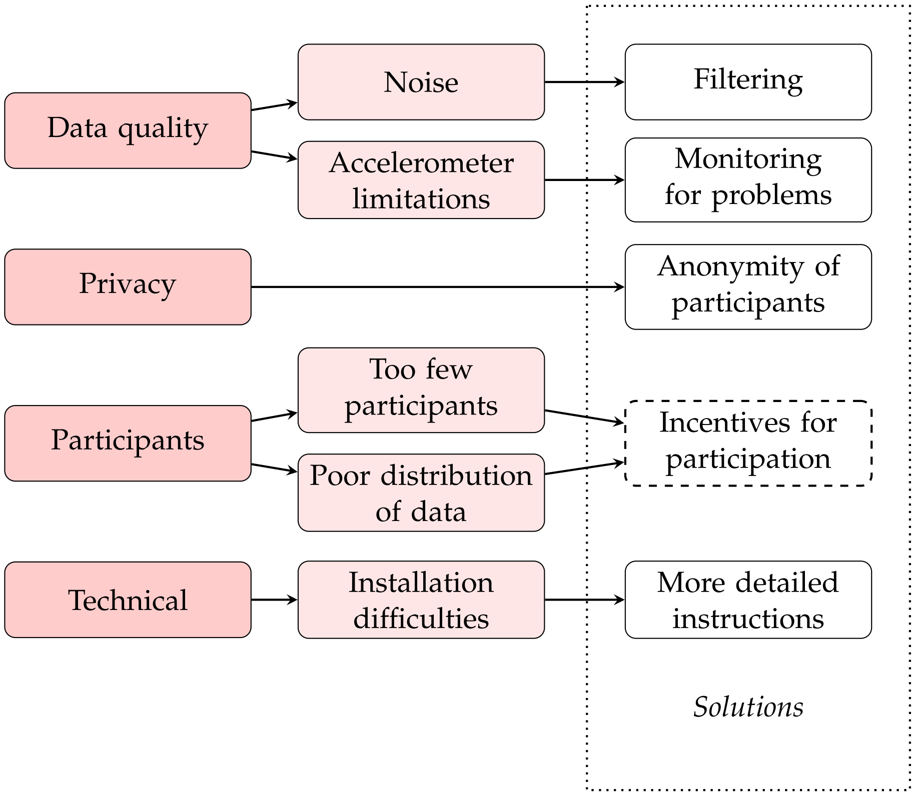


Figure 3.8: An overview of challenges related to data collection. Dashed line indicates that the problem needs more work in order to be solved.

Recruitment and participation of users

One goal of making a classifier is achieving a high level of generalization. In order to reach this goal, a representative selection of participants should be chosen, and they should each submit a high number of recordings distributed amongst all the available classes. In practice, this has not been possible to achieve.

First, the pool of possible participants was chosen from the author's list of contacts. A large number of people were asked, and some of them agreed to participate in the study[7]. However, only a few used the app for a significant amount of time, and most of the participants submitted very few recordings. This is unfortunate for the training of the classifier, as a higher number of recordings would make it easier to generalize. This is

[7] See section 4.1 for more details.

especially apparent for the less common classes, i.e. the emotional states that are less frequently experienced. We more often feel rather neutral than e.g. very activated or very unhappy.

Technical issues

Battery usage must be considered when developing applications that always run in the background. The recording of data and subsequent user interaction involve a significant amount of work and therefore drain the phone rather rapidly. However, this lasts for a very short time. At any other time, the application only monitors the accelerometer once per second, which requires very little resources. Therefore, there have not been any issues with participants' phones running out of battery at an excessive rate[8]

While the participants received clear instructions on how to install the application, i.e. opening an .apk file after allowing their phone to install applications in such a manner, there were still reports of difficulties related to installation. One such instance was resolved by sending the user a new version of the application, while another user had to disable another application in order to be allowed to perform the installation. While neither of these issues were highly problematic, it seems like a good idea to make even more detailed instructions.

There has also been one instance of a corrupted recording. In that situation, the recording was stored incompletely in the database, i.e. only about 70% of the expected lines of data were stored, and the recording stopped in the middle of a value. This was detected when reading data in advance of classification. The application does not submit data before the recording is finished properly, so the error must have happened in relation to sending or storing the data. Exactly where it happened is however unknown, as there are no logs where it is mentioned, and it happened only once. For future reference it would be a good idea to add some kind of data validation to the server, in order to prevent corrupt data from being stored, even though the reading of data prior to classification handles such issues.

3.3 Preprocessing of data

3.3.1 Sensor data

The accelerometer data presents some difficulties when it comes to data quality:

- The accelerometer is not perfectly accurate, and this adds some noise.
- When walking with the phone in a pocket, it does not have a consistent positioning, so it is difficult to compare readings from individual axes.

[8] One user reported such problems, but the issue was fixed in a later version of the application before the bulk of participants were recruited.

- The location of the pocket has an impact on the readings, i.e. lying in a pocket located further down on the leg might cause the phone to experience more jerk than lying in a pocket further up.
- Particular gait characteristics of the participant will influence the recordings.
- There is much uncertainly as to whether the participant has been moving for the entire duration of the recording.

Some of these problems will be addressed during feature extraction, see section 3.4. The issue of whether the participant actually has been moving, however, makes a significant impact on the results if we try to extract features directly from the raw data. As will be described in the first experiments in section 4.2, it is necessary to extract the periods of actual motion before extracting features. This becomes evident by comparing the two examples of accelerometer data recordings in figure 3.9.

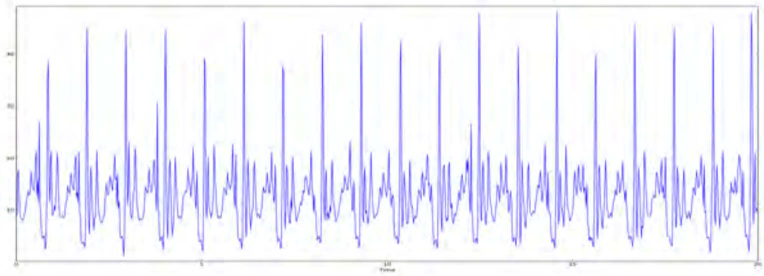
Figure 3.9a shows a recording where the participant most likely has been walking for the entire duration of the recording. We can see the characteristic peaks with a frequency of approximately 1Hz, which is the normal walking cadence. In this case it can be reasonable to calculate certain features based on the entire recording, e.g. features related to the total acceleration, jerk or periodicity of the movements.

However, the second recording in Figure 3.9b clearly does not have the same characteristics as the first. It looks like the participant might have been walking between the 2. and 9. second, but it is difficult to say what happened for the remainder of the recording. What matters is that only periods of movement are interesting to include for feature extraction in this work. Therefore, the accelerometer data has to be segmented into steps before features can be extracted.

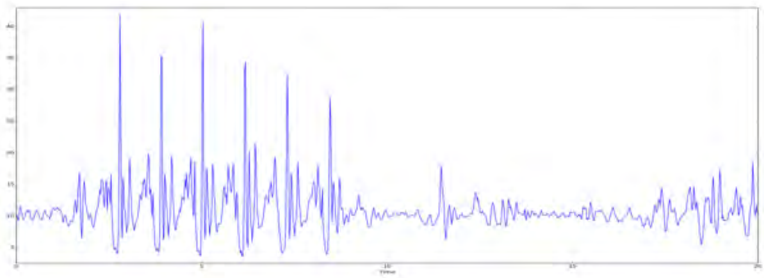
Noise reduction: Moving average filter

In order to reduce the noise, the acceleration data is first filtered with a *moving average filter*. Like the name indicates, for each recording, this filter is applied to each of the three raw acceleration vectors $\vec{a_x}, \vec{a_y}$ and $\vec{a_z}$, calculating each new value as the average of itself and the values directly adjacent to it. This happens before any other processing is performed. The number of adjacent values involved in each calculation can vary, but moving one or two values in each direction is normal, as used by e.g. Cui et al [16]. This gives the following filters:

$$m_3 = \left\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right\}$$
$$m_5 = \left\{\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right\}$$



(a) A recording where the participant has been walking all the time.



(b) A recording where the participant does not seem to have been walking all the time.

Figure 3.9: The plots show the length of the acceleration vector from two different recordings.

These filters can also be weighted, which is useful when only a smaller amount of smoothing is desired, e.g:

$$w_3 = \left\{ \frac{1}{4}, \frac{2}{4}, \frac{1}{4} \right\}$$
$$w_5 = \left\{ \frac{1}{16}, \frac{4}{16}, \frac{6}{16}, \frac{4}{16}, \frac{1}{16} \right\}$$

All these filters have been tried for the data involved in this work, and they have resulted in improvement of classification at some point. m_3 has been used for the final experiments in chapter 4.

Step segmentation

A step is defined as the movement that happens between two consecutive times of the same foot hitting the ground during walking, as explained in section 2.3. Figure 3.10 illustrates the stages of the step segmentation. For technical details, see section A.3.1. An example of how the step segmentation works is shown in Figure 3.11, based on the recording shown in Figure 3.9b:

- Figure 3.11a shows the absolute value of the derivative of the acceleration, i.e. the *jerk* of the movement. In practice, using the jerk seems to work better than using the acceleration for this way of segmenting steps.

- Figure 3.11b is the result of the jerk values being submit to two kinds of filtering processes:

  1. A 5. order digital Butterworth filter with a critical frequency of 0.05. This is a strong low-pass filter that removes all the smaller variations.
  2. Then all the values below a certain threshold are set to zero.

  We can then find the maxima of this plot, and this is used as an indication of possible steps. In order to increase the likelihood of actual walking having taken place, a requirement of a certain amount (here, 4, chosen as a reasonable assumption) of consecutive maxima, each separated from its neighbours by a distance indicative of reasonably normal walking cadence, is set. This causes us to end up with zero or more periods of assumed motion from each recording.

- These resulting periods of motion are then split into steps based on the location of the jerk maxima. Figure 3.11c shows the steps when the boundaries are placed on the maxima themselves. This has given certain undesirable outcomes where some steps contain two acceleration maxima, while others contain none[9]. Therefore, the split is rather performed in the middle of two maxima, and the limits for the two outer steps in a period of motion are decided through interpolation. This results in the graph in Figure 3.11d.

The success of the step segmentation is evaluated by visual inspection of plots of the data sets. Looking at plots of each recording in turn, we can judge whether the segmentation is reasonable. When tuning the parameters, the result should be rather strict, i.e. it is better to leave out (parts of) a recording that could have been included, than including (parts of) a recording that should have been left out. Looking through the entire data set when using the final chosen set of parameters, only one of the included recordings contained parts that definitely should have been left out. Significantly more recordings were left out unnecessarily, but this number was still reasonably low (<10%).

Upon completion of the preprocessing, there are three different levels from which features can be extracted:

1. The entire recording. This is not very desirable, as there might be parts of the recording where there has not been proper movement.
2. Extracted periods of motion. As mentioned above, such periods consist of a minimum of four steps, but might last for approximately the entire recording. Within these periods of motion we assume that the movement consists of actual steps. The algorithm should be tuned

[9] The acceleration maxima are not located in exactly the same place as the jerk maxima

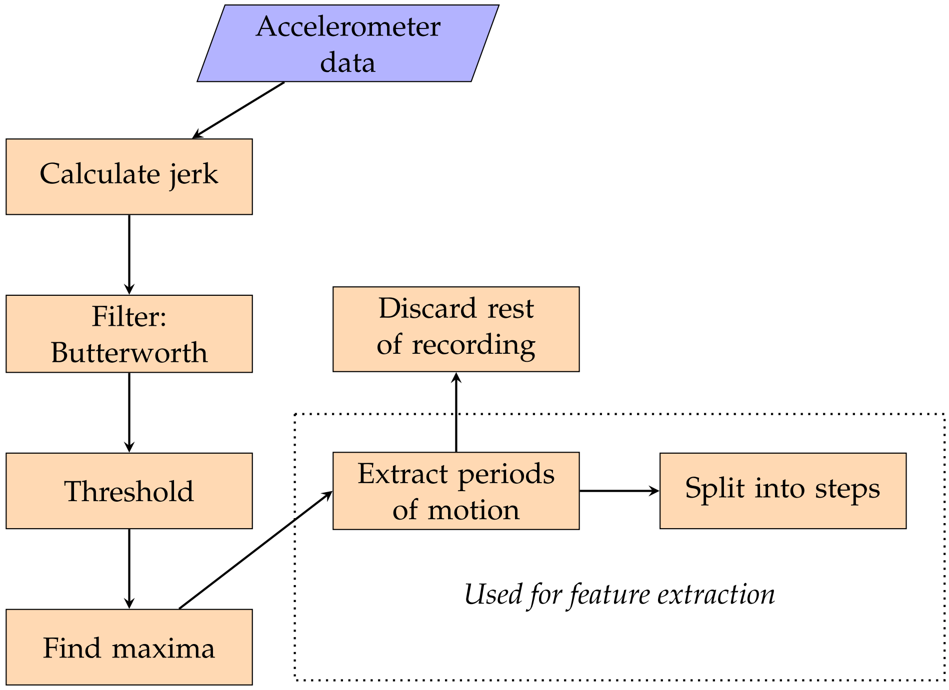


Figure 3.10: Overview of the step segmentation.

to discard data where there is reasonable doubt as to whether that assumption is correct.

3. Individual steps. Here the interpolated step limits are used to enable extraction of step-based features.

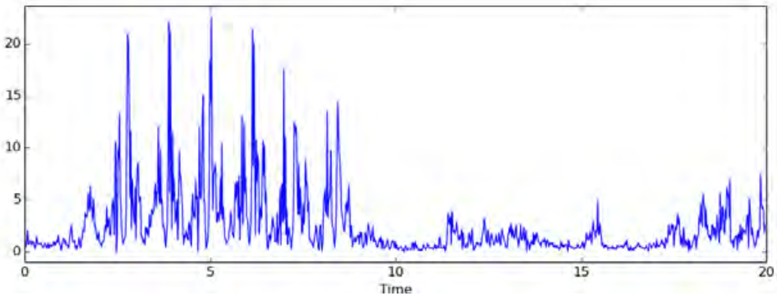
For later feature extraction, the periods of motion and the individual steps will be used as sources. See section A.3.1 for details of the implementation.

3.3.2 Self-reported emotional data

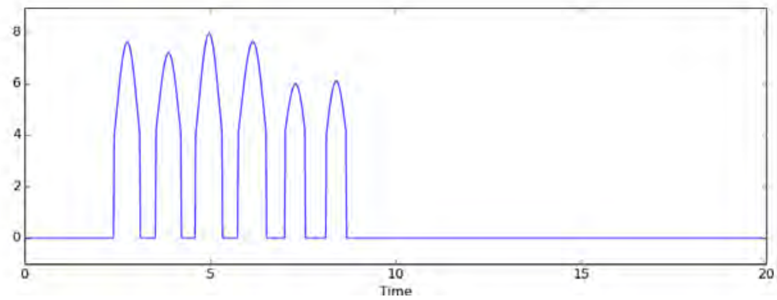
As shown in figure 3.2, the participants have five options to choose from in each dimension of emotions. While such a data resolution might be interesting, in order to increase the classification accuracy, the two highest options and the two lowest options will be combined in order to get three different classes on each scale: *low, medium and high*, see Table 3.1.

Table 3.1: The user-reported data on a 1-5 scale is converted to three classes.

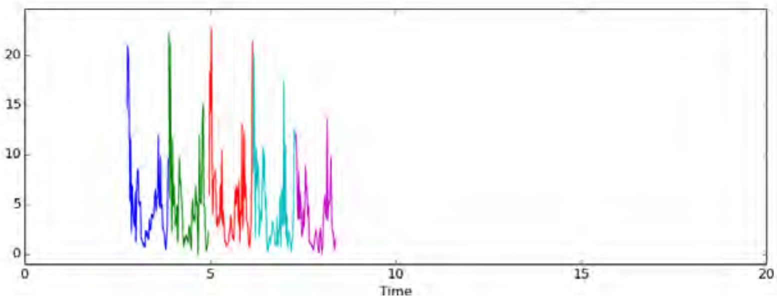
User-reported data	Class
0-1	Low
2	Medium
3-4	High



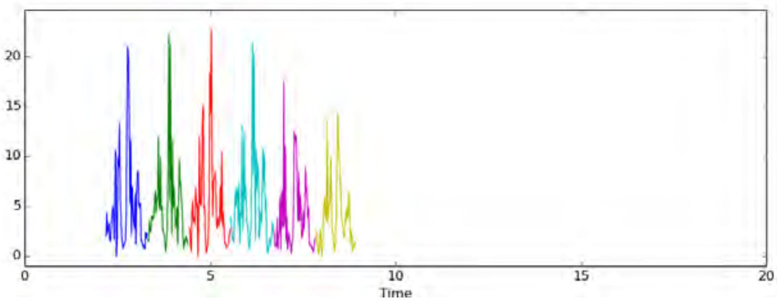
(a) The jerk of the movement.



(b) Jerk after Butterworth filtering and thresholding.



(c) Steps are segmented using the maxima from the previous figure as step limits.



(d) Steps are segmented using the point in the middle of two maxima as step limits.

Figure 3.11: The different stages of step segmentation.

3.4 Feature extraction

Choosing good features is essential for the success rate of the classifier. In this section a range of possible features will be listed, and in the next chapter the most promising ones will be chosen.

The sensors output their measurements on three separate axes. In this work we do not intend to take into account the position in which the phone is held, so it is reasonable to consider the total movement rather than the movement on the individual axes when calculating features. For each recording of M accelerometer readings $a_{x,i}, a_{y,i}, a_{z,i}$, beginning at time $t_0 = 0$ and ending at time t_{M-1}, we have the following values:

$$\vec{a_x} = \{a_{x,0}, a_{x,1}, \ldots, a_{x,M-1}\}$$
$$\vec{a_y} = \{a_{y,0}, a_{y,1}, \ldots, a_{y,M-1}\}$$
$$\vec{a_z} = \{a_{z,0}, a_{z,1}, \ldots, a_{z,M-1}\}$$

These values are the basis for the calculation of features, and in this section we assume that the data has been segmented into steps as described in section 3.3.1. This means that the values in the acceleration vectors $\vec{a_x}, \vec{a_y}$ and $\vec{a_z}$ consist only of the parts of the original recording where we have found that the participant has been moving, i.e. the location of the detected steps.

The choice of features is based mainly on what has been attempted in related projects, e.g. [1, 5, 8, 14–16, 23, 25, 53, 77], most of which are mentioned in section 2.6. While the other reviewed studies generally classify emotions in other ways than (only) through accelerometer data [1, 15], classify activities rather than emotions [8, 25, 53] or collect data from stylized[10] expression of emotions or in ways that do not specifically involve walking [5, 6, 14, 23], their methods of feature extraction are relevant also for this project.

Below, the suggested features will be marked with gray.

Acceleration

$$\text{For } i = 0 \ldots M-1:$$
$$a_i = \sqrt{a_{x,i}^2 + a_{y,i}^2 + a_{z,i}^2}$$

giving the acceleration vector

$$\vec{a} = \{a_0, a_1, \ldots, a_{M-1}\}$$

[10]Where emotions are expressed explicitly, i.e. the participants are told to express certain emotions through their actions, as opposed to this work, where emotions are implicitly expressed through normal walking

Mean acceleration:

$$\bar{a} = \frac{1}{M} \sum_{i=0}^{M-1} a_i \tag{3.1}$$

The standard deviation of a set of values tells us something about the distribution of those values and can be interesting to include as a feature.

Standard deviation of acceleration:

$$\sigma_a = \sqrt{\frac{1}{M} \sum_{i=0}^{M-1} (a_i - \bar{a})^2} \tag{3.2}$$

Similarly, the peak acceleration for each step can be calculated, giving a measure of the strongest movement in a step. With s steps and the peak acceleration for step k denoted as $a_{top,k}$, a measure for the mean peak acceleration $\bar{a}_{top}$ can be calculated as follows:

$$\bar{a}_{top} = \frac{1}{s} \sum_{k=0}^{s-1} a_{top,k} \tag{3.3}$$

And the standard deviation:

$$\sigma_{a_{top}} = \sqrt{\frac{1}{s} \sum_{k=0}^{s-1} (a_{top,k} - \bar{a}_{top})^2} \tag{3.4}$$

Another measure of the energy in the acceleration is the root of the sum of the squared acceleration values. Compared to the mean acceleration, this feature lets the highest values give a larger contribution.

$$\bar{a}_{rss} = \sqrt{\frac{1}{M} \sum_{i=0}^{M-1} a_i^2} \tag{3.5}$$

Jerk

Jerk is the first derivative of acceleration.

$$\begin{aligned}
&\text{For } i = 0 \ldots M-2: \\
&\quad j_{x,i} = a'_x(t) \approx \frac{a_{x,i+1} - a_{x,i}}{\Delta t} \\
&\quad j_{y,i} = a'_y(t) \approx \frac{a_{y,i+1} - a_{y,i}}{\Delta t} \\
&\quad j_{z,i} = a'_z(t) \approx \frac{a_{z,i+1} - a_{z,i}}{\Delta t} \\
&\quad j_i = \sqrt{j_{x,i}^2 + j_{y,i}^2 + j_{z,i}^2}
\end{aligned}$$

giving the jerk vector

$$\vec{j} = \{j_0, j_1, \ldots, j_{M-2}\}$$

Mean jerk:

$$\bar{j} = \frac{1}{M-1} \sum_{i=0}^{M-2} j_i \tag{3.6}$$

Standard deviation of jerk:

$$\sigma_j = \sqrt{\frac{1}{M-1} \sum_{i=0}^{M-2} (j_i - \bar{j}_i)^2} \tag{3.7}$$

Similarly, the peak jerk for each step can be calculated. With s steps and the peak jerk for step k denoted as $j_{top,k}$, a measure for the mean peak acceleration $\bar{j}_{top}$ can be calculated as follows:

$$\bar{j}_{top} = \frac{1}{s} \sum_{k=0}^{s-1} j_{top,k} \tag{3.8}$$

And the standard deviation:

$$\sigma_{j_{top}} = \sqrt{\frac{1}{s} \sum_{k=0}^{s-1} (j_{top,k} - \bar{j}_{top})^2} \tag{3.9}$$

The modified mean, calculated as the root of the sum of squared jerk values:

$$\bar{j}_{rss} = \sqrt{\frac{1}{M-1} \sum_{i=0}^{M-2} j_i^2} \tag{3.10}$$

The mean and standard deviation of acceleration and jerk values have been calculated based on the entire segments of motion. Modifying this process to calculate these values for each step, and then taking the mean of the resulting values, has shown slightly better performance in some cases. This alternate approach gives equal contribution from each step rather than from each data point.

In order to perform this modified calculation, start with a recording consisting of s steps. Each step consists of p_k data points, where $k \in \{0, \ldots, s-1\}$. The number of data points is different from step to step. Denoting the start index for each step $start_k$, the mean step-based acceleration can be calculated as

$$\bar{a}_{step} = \frac{1}{s} \sum_{k=0}^{s-1} \left(\frac{1}{p_k} \sum_{i=start_k}^{start_k+p_k-1} a_i \right)$$

Similarly, the standard deviation of the acceleration and the mean and standard deviation of the jerk can be calculated based on steps.

The difference between the two methods is illustrated in Figure 3.12.

Step duration

When calculating the steps in section 3.3.1, step limits are defined. These data points can be used to calculate the step duration, as the total time for each recording and the number of data points in each recording is known. With s steps, each consisting of p_k data points, a total recording time of T, and M data points in a recording, the mean step duration is

$$\bar{t}_s = \frac{1}{s} \sum_{k=0}^{s-1} \frac{p_k T}{M} \tag{3.11}$$

and the standard deviation

$$\sigma_{t_s} = \sqrt{\frac{1}{s} \sum_{k=0}^{s-1} \left(\frac{p_k T}{M} - \bar{t}_s \right)^2} \tag{3.12}$$

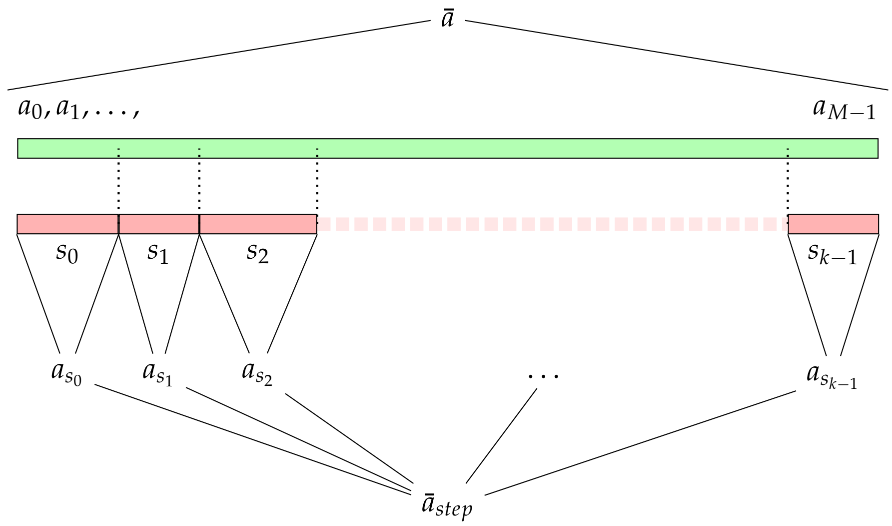


Figure 3.12: In the top of the picture the mean acceleration is calculated based on the entire period of motion extracted from a recording. Below, the period of motion is split into steps, and the mean acceleration is calculated from intermediate values based on each step.

Fourier Transform

The Fourier Transform transforms a time series signal into the frequency domain [22]. This is particularly relevant for data where there is some kind of periodicity, which is true for the accelerometer data recorded in this project. The Fourier Transform of the acceleration is calculated by the *Fast Fourier Transform* [21] algorithm and based on the following equation:

$$A_k = \sum_{m=0}^{M-1} a_m e^{-i2\pi km/M}$$

where $k \in 0, \ldots, M-1$.

The resulting Fourier coefficients A_k, or a subset thereof, can be used as features.

Skewness and kurtosis

Skewness [67] and kurtosis [36] are statistical measures of the asymmetry of the shape of a distribution . For a set of values A, skewness is calculated as follows:

$$Skew(A) = E\left[\left(\frac{A-\mu}{\sigma}\right)^3\right]$$

This feature will be calculated for each step, similar to the alternative calculation of the mean of the acceleration and jerk features above. With each step consisting of p_k data points, and the start index for each step denoted $start_k$, the step-based measure of skewness can be calculated as

$$Skew(A)_{step} = \frac{1}{s}\sum_{k=0}^{s-1}\left(\frac{\frac{1}{p_k}\sum_{i=start_k}^{start_k+p_k-1}(a_i-\bar{a})^3}{\left(\frac{1}{p_k}\sum_{i=start_k}^{start_k+p_k-1}(a_i-\bar{a})^2\right)^{\frac{3}{2}}}\right) \tag{3.13}$$

Similarly, kurtosis is calculated as

$$Kurt(A)_{step} = \frac{1}{s}\sum_{k=0}^{s-1}\left(\frac{\frac{1}{p_k}\sum_{i=start_k}^{start_k+p_k-1}(a_i-\bar{a})^4}{\left(\frac{1}{p_k}\sum_{i=start_k}^{start_k+p_k-1}(a_i-\bar{a})^2\right)^{2}}\right) \tag{3.14}$$

Power Spectral Density (PSD)

PSD is a measure of the frequency distribution of power of a time series signal, where power is measured by the square of the values of the time series [51]. Utilizing the Fourier Transform, the discrete version of PSD is calculated as follows:

$$\tilde{S}_{aa}(\omega) = \frac{(\Delta t)^2}{T}\left|\sum_{j=1}^{M} x_j e^{-i\omega j}\right|^2$$

From the resulting values we can calculate the mean

$$\bar{S}_{aa} = \frac{1}{l}\sum_{k=0}^{l-1}\tilde{S}_{aa,k} \tag{3.15}$$

where l is the length of the $\tilde{S}_{aa}$ array, and the standard deviation

$$\sigma_{S_{aa}} = \sqrt{\frac{1}{l}\sum_{k=0}^{l-1}(\tilde{S}_{aa,k} - \bar{S}_{aa})^2} \tag{3.16}$$

3.4.1 Feature standardization

In order to improve classifier accuracy, some kind of normalization/standardization of features is recommended. One way to do this is to subtract the mean from each feature value and then divide by the standard deviation.

With N features and M data entries we get an MxN matrix of feature values. First, for each feature $f_{i,e}$, where $i \in \{1, \dots, N\}$ and $e \in \{1, \dots, M\}$, we find the distribution mean $\bar{f}_i$ and standard deviation σ_i:

$$\bar{f}_i = \frac{1}{M}\sum_{e=1}^{M} f_{i,e}$$

$$\sigma_i = \sqrt{\frac{1}{M}\sum_{e=1}^{M}(f_{i,e} - \bar{f}_i)^2}$$

Then each feature can be standardized:

$$\hat{f}_{i,e} = \frac{f_{i,e} - \bar{f}_i}{\sigma_i} \tag{3.17}$$

This gives us a distribution of features with means of zero and standard deviations of one, allowing the various features to have equal impact on the classification.

Alternatively, the maximum and minimum feature values can be used instead of the mean and standard deviation:

$$\hat{f}_{i,e} = \frac{f_{i,e} - \min\limits_{0 \le j < M}(f_{i,j})}{\max\limits_{0 \le j < M}(f_{i,j}) - \min\limits_{0 \le j < M}(f_{i,j})} \tag{3.18}$$

In this case the feature values will end up in the interval $[0, 1]$.

3.4.2 Individual movement bias

The goal of the experiments performed in this thesis is to find a relation between a person's emotional state and the way the person moves as recorded by a smartphone carried in a pocket. A possible source of classification error is a person's *individual movement bias*, the characteristics of a person that might affect the movements in all emotional states. If we e.g. decide to use a feature which is based on the acceleration or total movement of the phone, the results might be entirely different for a person who generally uses very small movements compared to a person who generally uses larger movements.

A solution to this might be to calculate the *unbiased motion features*, as suggested by Bernhard et al. [5]. The unbiased motion features take into account that there might be certain tendencies influencing a person's general movements and try to correct for this by subtracting the personal bias $\bar{\varphi}_p$:

$$\hat{\varphi}_{p,m} = \varphi_{p,m} - \bar{\varphi}_p \tag{3.19}$$

for a person p and movement m.

This could allow us more easily to classify the movements based on their relative differences within the collection of samples from a specific person, as it seeks to reduce the impact of individual movement bias.

Estimating the movement bias $\bar{\varphi}_p$ can according to Bernhardt et al. be done in two ways:

- Suggesting a statistical value. The source of this value can be demographical data or other information.
- Using the average of a larger number of movements. This is a more accurate estimation, but requires a collection of movement samples.

Both these options are challenging, but in practice the latter one probably has higher potential for accuracy. We can use two approaches, one during training of the classifiers, and the other when the results are to be implemented in an application, e.g. as mentioned in section 1.2.

Firstly, the training is limited by the nature of the training data. It consists of samples from a selection of users where some have submitted a large number of samples, while others only have submitted a few. A small number of samples means that we get a low reliability when calculating a personal bias value. However, it might still be better than ignoring the bias altogether, so the feature values f will be adjusted based on the average values for each participant and feature:

$$\hat{f}_{p,i,j} = f_{p,i,j} - \bar{f}_{p,i} \tag{3.20}$$

for a participant p, feature number i and recording number j.

Secondly, once the training is complete, the classifier has to be able to handle input from new users. While results would be better if it were feasible to train the classifier on each person, the method has significantly higher potential for use if it works right away. Therefore, a statistical value, e.g. a value based on the average across all the training data, can be used as a starting point for bias adjustment for new users. Having established such a starting value, this value can be adjusted gradually as data from new users is recorded.

Chapter 4

Experiments

This chapter covers all the classification experiments performed and their results. The following sections will look at how different factors influence the results:

- Choice of classification algorithm
- Choice of features
- Selection of participants
- Sampling rate
- Individual movement bias

The presentation in this chapter is ordered based on how the work progressed. First comes information about the participants in section 4.1 and results from some of the earlier experiments in section 4.2. Then follows information about feature selection in section 4.3. Finally, the results of the classification will be presented in section 4.4

4.1 Data collection: Participants

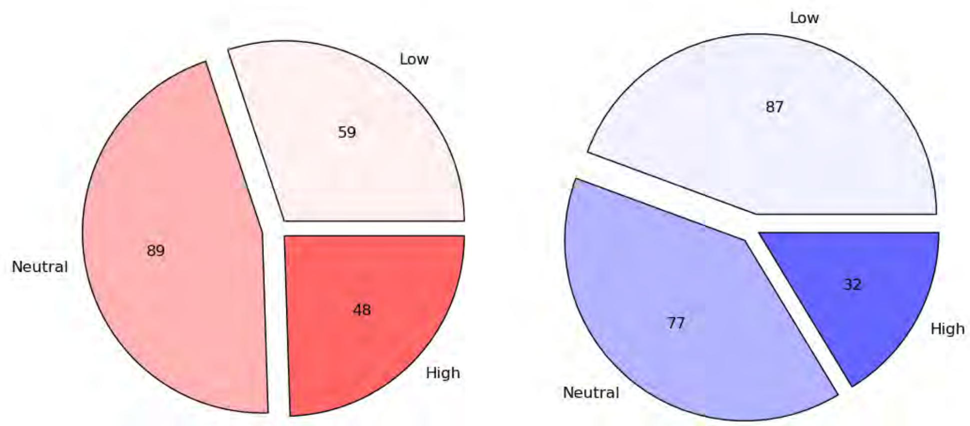
Participants for the data collection were found through the author's personal network. Out of 62 people asked, 22 accepted. The remainder either did not respond or did not use an Android phone.

Those 22 participants received instructions together with the application *Emotions* and confirmed that they were going to use it. However, only ten actually participated, and only three, the author of this thesis included, submitted more than 10 recordings. In total, 196[1] recordings were collected. Figure 4.1 shows the distribution of classes for each of the two dimensions in those recordings.

[1] The initial number is higher, but these are the ones remaining after preprocessing has filtered some out.

Table 4.1: Participants

# asked:	62
# accepted:	22
# participated	10
# recordings	196



(a) Pleasantness. (b) Arousal

Figure 4.1: Distribution of classes

4.2 First results: Two participants and two basic features

After using the app myself for a few weeks, I had collected a data set consisting of 62 recordings (approximately 1000 lines of data per recording, where each line is an accelerometer reading consisting of x-, y- and z-values, as explained in section 3.2.3). Training a classifier and testing on data from only one person is not desirable, as it greatly reduces the ability to generalize to other people. The data set is also rather small. However, it eliminates the need to take into consideration the differences between individuals' movement patterns, e.g. the individual movement bias (see section 3.4.2), and it is a simple way to start. The results from this classification are presented below.

A similar classification experiment was performed separately on data from one of the other participants, see section 4.2.2.

4.2.1 Participant #1

This classification was performed with the parameters listed in table 4.2, using only two basic features.

Table 4.2: Classification details

Classifier	Multilayer perceptron
# recordings	62
# participants	1
# runs of classifier, each using specified validation	1
Features	Average and standard deviation of the movement jerk (see section 3.4), no pre-processing.
Standardization of features	Rescaling, see equation 3.18.
Sampling rate	200 Hz
Validation	K-fold cross-validation as described in section A.3.1
Classes	Low (0-1), neutral (2) and high (3-4)

Results for the pleasantness dimension

Average result: 57.8%
Best result: 84.6%
Confusion matrix for the best result:

$$\left[\begin{array}{c|ccc} & Low & Neutral & High \\ \hline Low & 0 & 0 & 0 \\ Neutral & 1 & 8 & 1 \\ High & 0 & 0 & 3 \end{array}\right]$$

The confusion matrix shows the distribution of classification outputs and is explained in section 2.4. We see that a large proportion of the data set belongs the neutral class, and in the iteration that yielded the best result, both the neutral and high classes were mostly correctly classified. The average is however much lower, and later sections will show that the pleasantness dimension is very hard to classify correctly based on accelerometer data.

Results for the arousal dimension

Average result: 75.5%
Best result: 91.7%
Confusion matrix for the best result:

$$\left[\begin{array}{c|ccc} & Low & Neutral & High \\ \hline Low & 1 & 0 & 0 \\ Neutral & 0 & 9 & 0 \\ High & 0 & 1 & 1 \end{array}\right]$$

Also for the arousal dimension most of the samples belong to the neutral class, but in this case the average result is much higher. This is as expected, as the chosen features are a measure of the energy of the movement and, as can be seen later, predicative of the arousal dimension.

4.2.2 Participant #2

This classification used similar parameters as the previous section, and the details can be seen in table 4.3.

Table 4.3: Classification details

Classifier	Multilayer perceptron
# recordings	43
# participants	1
# runs of classifier, each using specified validation	1
Features	Average and standard deviation of the movement jerk (see section 3.4), no pre-processing.
Standardization of features	Rescaling, see equation 3.18.
Sampling rate	200 Hz
Validation	K-fold cross-validation as described in section A.3.1
Classes	Low (0-1), neutral (2) and high (3-4)

Results for the pleasantness dimension

Average result: 39.0%
Best result: 75.0%
Confusion matrix for the best result:

$$\left[\begin{array}{c|ccc} & Low & Neutral & High \\ \hline Low & 0 & 1 & 0 \\ Neutral & 1 & 5 & 0 \\ High & 0 & 0 & 1 \end{array}\right]$$

Results for the arousal dimension

Average result: 62.9%
Best result: 77.8%
Confusion matrix for the best result:

$$\left[\begin{array}{c|ccc} & Low & Neutral & High \\ \hline Low & 4 & 1 & 1 \\ Neutral & 0 & 3 & 0 \\ High & 0 & 0 & 0 \end{array}\right]$$

4.2.3 Discussion

The data sets for both classifications were clearly too small. By looking at the confusion matrices, we can see that some classes are much higher represented than others, which is no surprise, as we tend to be in a neutral emotional state more often than not. However, that means that the we need a higher number of recordings in order to get a sufficient amount of samples from the non-neutral states.

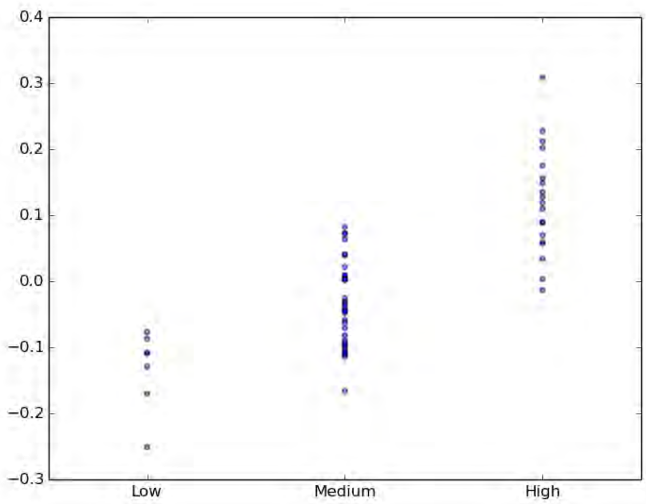
We see that the classification performed on the data I recorded myself worked much better than the one performed on data from one of the other participants. This can easily be understood by comparing scatter plots of the average jerk – one of the features used – as a function of the reported arousal, see figure 4.2.

While the first plot (4.2a) shows a clear trend, the other (4.2b) does not show a particularly discriminatory feature. One of the reasons for this difference might be that I knew exactly how the data collection worked and ensured that all the recordings I submitted were made in situations where I actually had been walking all the time. This assumption cannot be applied to other participants. The app tries to record only when the participant is walking, and also asks for a confirmation before the recording is sent to the server, but this does not guarantee that the entire period of a recording is made when a person is walking. Therefore, more preprocessing of the data might be necessary in order to filter out segments where the participant has been doing something else. A method for accomplishing this is suggested in section 3.3.

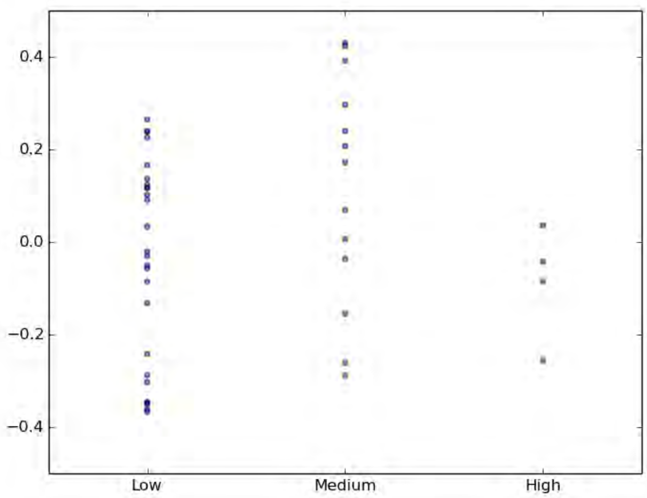
In the following section, the same data has been used, but instead of extracting features based on the entire recording, the step segmentation described in section 3.3 is performed first. As a result, some data entries are eliminated, as an insufficient number of steps are found. This probably means that the participant has not been walking during the recording, and such data only adds noise to the classification. Therefore we will see better results from this classification.

4.2.4 Participant #2: With features calculated after step segmentation

Table 4.4 shows that the parameters are the same as the previous run, but features are extracted from periods of motion rather than from the entire recording. Seven recordings have been excluded because the step segmentation algorithm was unable to extract a sufficiently long period of walking from those recordings. While the accuracy for the arousal dimension has improved, there is no difference for the pleasantness dimension. The improvement seems reasonable when looking at the scatter plot of the normalized feature values in figure 4.3, which compared to the previous one is much more discriminatory between the low and medium levels of arousal, but not so much for the high. There are however very few samples of the latter.



(a) participant #1



(b) participant #2

Figure 4.2: Scatter plots of the normalized movement jerk feature

Results for the pleasantness dimension

Average result: 40.3%
Best result: 57.1%
Confusion matrix for the best result:

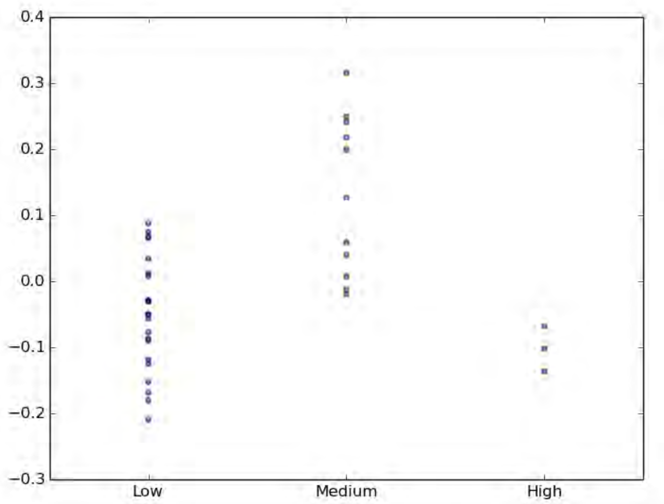


Figure 4.3: participant #2: Scatter plots of the normalized step-based movement jerk feature

	Low	*Neutral*	*High*
Low	1	1	0
Neutral	0	3	1
High	1	0	0

Results for the arousal dimension

Average result: 71.1%
Best result: 87.5%
Confusion matrix for the best result:

	Low	*Neutral*	*High*
Low	6	1	0
Neutral	0	1	0
High	0	0	0

The average result here is significantly better than the previous one, 71.1% vs 62.9%. Segmenting the accelerometer recordings into periods of motion before extracting features proves to be essential, and this preprocessing will be used for all further experiments.

4.3 Feature selection

A number of features have been described in the previous chapter. In this section they will be evaluated, and the ones that contribute to getting the best classification results will be used in the following sections. This feature

Table 4.4: Classification details

Classifier	Multilayer perceptron
# recordings	36
# participants	1
# runs of classifier, each using specified validation	1
Features Equations 3.6, 3.7	
Standardization of features	Rescaling, see equation 3.18.
Sampling rate	200 Hz
Validation	K-fold cross-validation as described in section A.3.1
Classes	Low (0-1), neutral (2) and high (3-4)

selection is first performed as explained in section 2.4. For this purpose, the following parameters, the choice of which make an impact on the resulting feature set, have been used when running the classifier:

- Classifier: Support vector machine
- Participants: Only the ones that have registered at least ten recordings
- Individual movement bias: No

Running the RFE algorithm gives us two sets of features; one used for the pleasantness dimension and one for the arousal dimension, viz. equation 3.16 (pleasantness) and equations 3.1, 3.2, 3.4, 3.6, 3.11, 3.13, 3.14 (arousal). See table 4.5 for an overview and ranking of features according to the RFE algorithm. The ranking is included only for the arousal dimension, as the performance for pleasantness is generally low.

Ending up with just one feature for the pleasantness dimension is a consequence of the generally very low correlation between either of the features and the self-reported pleasantness. This indicates that the level of pleasantness is difficult to measure the way it is performed in this work. See section B.1 for plots of the rescaled feature values used for the feature selection, showing the distribution of the feature values plotted against the self-reported pleasantness and arousal, respectively. The plots have a very similar distribution for the different classes of pleasantness, showing that these features are not very useful for classifying the pleasantness dimension.

It is also worth pointing out that the resulting set of features depends on the choice of classification parameters. Other classifiers, different selection of participants etc. could cause the algorithm to output different features. It must also be mentioned that the RFE algorithm is *greedy*, i.e. it always chooses the optimal local solution (the best feature in each iteration). This means that an entirely different combination of features could work just as well or better, but it will not be discovered if it lies outside of the scope of always choosing the local optimum. An example is the feature set of

Table 4.5: An overview of the various features.

Name	Equation	Ranking (arousal)	Features used (A/P)
Mean acceleration	3.1[6]	7	A
Std acceleration	3.2[6]	6	A
Mean peak accel.	3.3	10	
Std peak accel.	3.4	4	A
Mean jerk	3.6[6]	3	A
Std jerk	3.7[6]	12	
Mean peak jerk	3.8	9	
Std peak jerk	3.9	13	
Mean step duration	3.11	1	A
Std step duration	3.12	14	
Skewness	3.13	5	A
Kurtosis	3.14	2	A
Mean PSD	3.15	8	
Std PSD	3.16	11	P

equations 3.1,3.2,3.11,3.15, which in some circumstances gives as good or possibly better classification results for the arousal dimension as the previous set described.

4.4 Classification results

This section contains the results of the classification. First, only the mean and standard deviation of acceleration are used as features, and recordings from all users are included. Classification using only those features establishes a base case where we only look at the energy of the movement. Intuitively, the energy should be connected to a person's state of activation/arousal, and therefore the accuracy for the arousal dimension is decent. In the following parts of this section, different potential improvements will be introduced:

- A more complex feature model, including all the features selected in section 4.3.
- Reducing the sampling rate of recordings. This is not expected to improve classification, but it reduces the requirements for storage and transfer of data.
- Selecting only participants with at least ten recordings.
- Individual movement bias, as explained in section 3.4.2

Each subsection will present the classification details and results, and table 4.6 shows some information that is common to all runs of the

[6] The step-based and squared variants of this feature were also evaluated, but did not improve the classification

classifiers, unless otherwise specified. Table 4.7 shows an overview of the different experiments.

Table 4.6: Common classification details

Classifiers	Decision tree (tree), support vector machine (svm) and multilayer perceptron (mlp)
# runs of classifier	200 (tree and svm) / 10 (mlp)
Standardization of features	Rescaling, see equation 3.18.
Sampling rate	50 Hz
Validation	K-fold cross-validation as described in section A.3.1
Comparison to random guess	With M recordings total and m recordings from the highest occurring class: $\frac{m}{M}$

Table 4.7: Overview of the experiments and best results

	Pleasantness		Arousal	
Experiment	Accuracy	Classifier	Accuracy	Classifier
Base case	46.1%	SVM	70.5%	MLP
+ selected features	48.1%	MLP	73.2%	SVM
+ selected users	50.9%	MLP	75.0%	SVM
+ ind. movement bias	48.2%	tree	75.3%	SVM

4.4.1 Simple feature model, normal sampling rate, all participants

See table 4.8 for classification details.

Table 4.8: Classification details, see table 4.6 for more info.

# recordings	196
# participants	10
Features	Equations 3.1, 3.2

Decision tree – Results for the pleasantness dimension

Average result: **45.9%** (random guess: 45.4%
Standard deviation: 2.8
Class distribution: Low (0-1): 59, neutral (2): 89 and high (3-4): 48

Decision tree – Results for the arousal dimension

Average result: **59.9%** (random guess: 44.4%
Standard deviation: 2.3
Class distribution: Low (0-1): 87, neutral (2): 77 and high (3-4): 32

Support vector machine – Results for the pleasantness dimension

Average result: **46.1%** (random guess: 45.4%
Standard deviation: 1.4
Class distribution: Low (0-1): 59, neutral (2): 89 and high (3-4): 48

Support vector machine – Results for the arousal dimension

Average result: **64.9%** (random guess: 44.4%)
Standard deviation: 2.1
Class distribution: Low (0-1): 87, neutral (2): 77 and high (3-4): 32

Multi-layer perceptron – Results for the pleasantness dimension

Average result: **44.9%** (random guess: 45.4%
Standard deviation: 1.1
Class distribution: Low (0-1): 59, neutral (2): 89 and high (3-4): 48

Multi-layer perceptron – Results for the arousal dimension

Average result: **70.5%** (random guess: 44.4%)
Standard deviation: 0.8
Class distribution: Low (0-1): 87, neutral (2): 77 and high (3-4): 32

Classifying the pleasantness dimension gives approximately the same results as performing a random guess. For arousal the results are decent, especially for the MLP. This indicates that much of the information needed to determine a person's state of activation from accelerometer data can be found by using a simple threshold calculated from the mean and standard deviation of the data.

4.4.2 All selected features, normal sampling rate, all participants

See table 4.9 for classification details.

Decision tree – Results for the pleasantness dimension

Average result: **42.5%** (random guess: 45.4%)
Standard deviation: 2.4

Table 4.9: Classification details, see table 4.6 for more info.

# recordings	196
# participants	10
Features	See table 4.5

Class distribution: Low (0-1): 59, neutral (2): 89 and high (3-4): 48

Decision tree – Results for the arousal dimension

Average result: **67.4%** (random guess: 44.4%)
Standard deviation: 2.5
Average confusion matrix:

$$\begin{bmatrix} & \mathit{Low} & \mathit{Neutral} & \mathit{High} \\ \hline \mathit{Low} & 75.2 & 17.9 & 21.4 \\ \mathit{Neutral} & 18.3 & 68.9 & 36.2 \\ \mathit{High} & 6.5 & 13.2 & 42.3 \end{bmatrix}$$

Class distribution: Low (0-1): 87, neutral (2): 77 and high (3-4): 32

Support vector machine – Results for the pleasantness dimension

Average result: **46.4%** (random guess: 45.4%
Standard deviation: 1.6
Class distribution: Low (0-1): 59, neutral (2): 89 and high (3-4): 48
Classification accuracy is higher using only this one feature than using all features listed in table 4.5.

Support vector machine – Results for the arousal dimension

Average result: **73.2%** (random guess: 44.4%)
Standard deviation: 1.3
Average confusion matrix:

$$\begin{bmatrix} & \mathit{Low} & \mathit{Neutral} & \mathit{High} \\ \hline \mathit{Low} & 83.1 & 21.7 & 32.2 \\ \mathit{Neutral} & 14.1 & 71.8 & 18.2 \\ \mathit{High} & 2.8 & 6.5 & 49.6 \end{bmatrix}$$

Class distribution: Low (0-1): 87, neutral (2): 77 and high (3-4): 32

Multi-layer perceptron – Results for the pleasantness dimension

Average result: **48.1%** (random guess: 45.4%
Standard deviation: 0.7

Class distribution: Low (0-1): 59, neutral (2): 89 and high (3-4): 48

Multi-layer perceptron – Results for the arousal dimension

Average result: **69.6%** (random guess: 44.4%)
Standard deviation: 1.6
Average confusion matrix:

$$\left[\begin{array}{c|ccc} & \mathit{Low} & \mathit{Neutral} & \mathit{High} \\ \hline \mathit{Low} & 77.8 & 22.5 & 25.9 \\ \mathit{Neutral} & 19.0 & 71.2 & 30.7 \\ \mathit{High} & 3.2 & 6.3 & 43.4 \end{array}\right]$$

Class distribution: Low (0-1): 87, neutral (2): 77 and high (3-4): 32

Now that the full selection of features is included, we see that the SVM classifier performs better than the MLP for the arousal dimension. The confusion matrices show that both methods are equally good at predicting the neutral class, while the SVM has higher accuracy for the low and high classes. The decision tree performs worse in all aspects. In general, the results are much better than they were for the simple feature model when it comes to arousal, while the decision tree actually performs worse for the pleasantness dimension. This shows that which features make up the the seemingly best feature set is different for the different classifiers.

4.4.3 All selected features, reduced sampling rate, all participants

As described in section 2.3, it has been shown that sampling frequencies down to 10Hz capture most of the information from the accelerometer in the context of activity recognition. Therefore, it would be interesting to examine the effects of reducing the sampling rate of the recorded data in this work.

Sampling frequencies of 25Hz, 16.7Hz and 12.5Hz were interpolated from the recorded data, which has a sampling frequency of 50Hz. This was done by selecting every 2., 3. and 4. line of each recording, respectively, and discarding the remaining lines. Other parameters involved in the preprocessing were adjusted, in order to ensure that approximately the same periods of motion and steps were identified.

The results of this experiment showed that reducing the sampling rate immediately had an impact on the classification accuracy. There can be different reasons for this result:

- Reducing the sampling rate might remove some of the information used for the classification, and the requirements for this kind of classification could potentially be higher than what it is for activity classification.
- The parameters related to preprocessing and classifiers could require more/better adjustment.

- Feature selection might give a different outcome when the sampling rate is reduced than it did for the normal sampling rate.

Further exploration of the importance of sampling rate has been deemed outside the scope of this work, but it would be interesting to examine it further in the future. For practical applications it is best to use as low sampling rate as possible, due to battery strain. It is also worth noting that using an even higher sampling rate than then one used in this work could potentially give better classification results.

4.4.4 All selected features, normal sampling rate, selected participants

See table 4.10 for classification details.

Table 4.10: Classification details, see table 4.6 for more info.

# recordings	186
# participants	3
Features	See table 4.5

Decision tree – Results for the pleasantness dimension

Average result: **46.5%** (random guess: 46.2%)
Standard deviation: 2.6
Class distribution: Low (0-1): 53, neutral (2): 86 and high (3-4): 47

Decision tree – Results for the arousal dimension

Average result: **67.5%** (random guess: 43.5%)
Standard deviation: 2.6
Class distribution: Low (0-1): 81, neutral (2): 73 and high (3-4): 32

Support vector machine – Results for the pleasantness dimension

Average result: **49.6%** (random guess: 46.2%)
Standard deviation: 1.9
Class distribution: Low (0-1): 53, neutral (2): 86 and high (3-4): 47

Support vector machine – Results for the arousal dimension

Average result: **75.0%** (random guess: 43.5%)
Standard deviation: 1.6

Average confusion matrix:

$$\begin{bmatrix} & \mathit{Low} & \mathit{Neutral} & \mathit{High} \\ \hline \mathit{Low} & 83.9 & 19.8 & 30.1 \\ \mathit{Neutral} & 14.9 & 73.4 & 17.5 \\ \mathit{High} & 1.2 & 6.8 & 52.4 \end{bmatrix}$$

Class distribution: Low (0-1): 81, neutral (2): 73 and high (3-4): 32

Multi-layer perceptron – Results for the pleasantness dimension

Average result: **50.9%** (random guess: 46.2%)
Standard deviation: 0.9
Class distribution: Low (0-1): 53, neutral (2): 86 and high (3-4): 47

Multi-layer perceptron – Results for the arousal dimension

Average result: **72.3%** (random guess: 43.5%)
Standard deviation: 1.2
Average confusion matrix:

$$\begin{bmatrix} & \mathit{Low} & \mathit{Neutral} & \mathit{High} \\ \hline \mathit{Low} & 79.8 & 18.9 & 24.7 \\ \mathit{Neutral} & 18.0 & 74.6 & 27.5 \\ \mathit{High} & 2.2 & 6.5 & 47.8 \end{bmatrix}$$

Class distribution: Low (0-1): 81, neutral (2): 73 and high (3-4): 32

Including only the participants who had submitted at least ten recordings each improved the results for both the SVM and MLP, while the decision tree still performs worse than the others. We see now that the results from the SVM and MLP are fairly similar, each having a slight lead on one of the dimensions. Now MLP is better at classifying the neutral class for arousal, while SVM still is more accurate for the low and high classes.

4.4.5 All selected features, normal sampling rate, selected participants, individual movement bias

See table 4.11 for classification details.

Table 4.11: Classification details, see table 4.6 for more info.

# recordings	186
# participants	3
Features	See table 4.5
Individual movement bias	Yes

Decision tree – Results for the pleasantness dimension

Average result: **48.2%** (random guess: 46.2%)
Standard deviation: 2.6
Class distribution: Low (0-1): 53, neutral (2): 86 and high (3-4): 47

Decision tree – Results for the arousal dimension

Average result: **70.6%** (random guess: 43.5%)
Standard deviation: 2.6
Class distribution: Low (0-1): 81, neutral (2): 73 and high (3-4): 32

Support vector machine – Results for the pleasantness dimension

Average result: **46.3%** (random guess: 46.2%)
Standard deviation: 1.3
Class distribution: Low (0-1): 53, neutral (2): 86 and high (3-4): 47

Support vector machine – Results for the arousal dimension

Average result: **75.3%** (random guess: 43.5%)
Standard deviation: 1.4
Class distribution: Low (0-1): 81, neutral (2): 73 and high (3-4): 32

Multi-layer perceptron – Results for the pleasantness dimension

Average result: **47.6%** (random guess: 46.2%)
Standard deviation: 1.1
Class distribution: Low (0-1): 53, neutral (2): 86 and high (3-4): 47

Multi-layer perceptron – Results for the arousal dimension

Average result: **73.1%** (random guess: 43.5%)
Standard deviation: 1.2
Class distribution: Low (0-1): 81, neutral (2): 73 and high (3-4): 32

Adjusting for individual movement bias was expected to improve the classification accuracy. This did however not seem to happen. While there are indications of a slight improvement for the arousal dimension, the pleasantness dimension got a reduced accuracy for two classifiers. Intuitively, such an individual adjustment should be able to improve the classification, but exactly how it is to be implemented needs further work before it can be used.

Chapter 5

Conclusion and future work

This chapter summarizes the results from the previous chapter and suggests improvements that can be made in order to get better results and make the methods more useful for practical applications.

5.1 Conclusion

This work intended to classify emotions over the two dimensions pleasantness and arousal, based on accelerometer data from a smartphone carried in the participants' pocket during natural walking. The results indicate that the arousal dimension can be predicted from such data, while the pleasantness dimension is much harder to predict. The best prediction rates achieved for arousal and pleasantness are 75.3% and 50.9%, respectively, for classifiers trained on data from all participants with at least ten contributions.

There are a number of elements which impact the resulting prediction rates:

- Preprocessing of data is essential, especially step segmentation where sections of data that do not consist of walking are removed. Using a moving average filter for noise reduction and reducing the number of classes for self-reported emotional data from 5 to 3 also has an impact on the results.
- A large amount of training data is required in order to get sufficient samples from the less common emotional states.
- The sampling frequency in this work was set to 50Hz. Lowering the sampling frequency on the recordings reduced the classification accuracy.
- Individual movement bias seems to be less important than initially expected. However, this needs further exploration before any conclusion can be made.
- Selection of features is generally important for the classification accuracy, although a lot of the information needed for predicting

the arousal dimension can be found directly from the average acceleration.

- The multilayer perceptron had the highest classification accuracy when using few features, while the support vector machine performed better with a more complex feature model. The decision tree consistently performed worse.

While there are many improvements to be made, the results indicate that a person's emotional state is to a certain extent reflected in the movements the phone is subject to while lying in the person's pocket when he/she is walking. This is a highly non-obtrusive way of detecting emotions, and it requires no extra equipment or interaction with the user. Therefore, the accelerometer-based method of emotion detection that is described in this thesis could be useful in practical applications. It can either be part of a more complex system for emotion detection, with a potential combined high accuracy, or be used alone, as a lower accuracy indicator of a person's emotional state.

5.2 Future work and applications

Some potential applications have been mentioned in the introduction, and they will not be explored further here. However, there are many opportunities to improve the classification results and adapt the methods to practical use.

Recruitment of more study participants

One of the main problems in this project is the low amount of participants. While more than 20 people agreed to participate, only a few participated to a meaningful degree, as explained in section 3.2.4. That makes it hard to give any predictions on this project's ability to generalize its results to persons outside of the group of participants. A future study would therefore need to ensure that a sufficient number of participants are recruited and motivated.

In retrospect, it is easy to see that further incentives for the participants would have been good in order to collect more data. Rewarding high participation is however difficult in this context, as we would not want the participants to actively try to make the app request data. This would counteract the idea of collecting data from a natural environment and potentially yield lower quality data. If the project were to be started over again, it would have been interesting to implement a way for the users to see statistics of how their responses had progressed over time. This is not a reward per se, and it does not encourage high activity or any kind of competition, but it enables the users to see some kind of result of their participation and adds a more self-centered reason for further participation. Even more functionality could be added, taking the users' statistics into consideration, but the work required for implementing such features is also a factor.

The classifier that has already been trained in this project could be used as a starting point for a later study. Initially using the existing parameters, the model would adapt to the individual participant over time and give predictions right from the start. This might be used as a way to illustrate to the user that the participation actually makes an impact on how the algorithm works.

Alternative ways to collect data

Different phone states were mentioned in section 3.2.1. In this work we have only considered the phone lying in the pocket while walking, but it would also be interesting to make further attempts on detecting emotions from the phone while it is being used. In that situation, the accelerometer would record the movements from the user's hand on the phone, and that can also possibly convey information about the user's emotional state.

More closely related to what has been done here is analyzing data from a phone that is carried other places than in a pocket, e.g. in a backpack. Such movements would also be periodic, and to some extent similar to the movement experienced when the phone is in a pocket. A future project could analyze data from such movement and find a way to detect whether the phone is in a backpack, a purse or a pocket and choose the appropriate parameters for step segmentation and other preprocessing and classification accordingly.

Repeating this work using a discrete model of emotions, as described in section 2.1.1, would also be interesting. The advantages of using a two-dimensional model have been discussed in section 3.1, and the results show promising prediction rates for the arousal dimension. However, it is hard to tell whether the advantages are high enough to warrant the increased difficulty of filling out the information in the app. Perhaps a discrete model would increase the validity of the self-reported data. This would in turn have potential for a more accurate model, even though the discrete model of emotions can be more difficult to implement in a good way.

Classification

There are many different methods for selecting features. The one that was chosen in this project is a reasonable approach, but it would be interesting to compare it to other methods. Possible alternatives worth investigating are e.g. principal component analysis and correlation-based feature selection.

In addition to features, the selection of classes could be modified. Currently, pleasantness and arousal are only viewed separately, but as discussed in section 2.1.2 they comprise the dimensions of the two-dimensional emotion space. In order to translate these results into the domain of discrete emotions, i.e. the common way of discussing emotions, the classifier will have to consider combinations of the two dimensions.

This is challenging as long as the classification results for pleasantness are so poor, however, but it should be further investigated. Collecting more data is probably necessary first, though.

At the same time, if a much higher number of participants were to be recruited, a measure of the significance of the results should be calculated. It is relevant to have an approximation of the classifier's ability to generalize.

Finally, the performance of the classifiers is highly dependent on their parameters. In this work, a lot of different parameters were tried, but for better performance it would be useful to further explore the optimization of classifier parameters.

Modifications for practical use

As mentioned in section 3.3.1, the step segmentation performed in this project was implemented rather strictly, i.e. it tends to leave out too many rather than too few recordings. This means that some times when a person is walking, the current algorithm will not classify it as such. While this choice is good for the training stage, where it is important not to get invalid training data, the requirements should be relaxed when implementing the results in some kind of application.

For better performance in practical use, it would be possible to implement a measure of likelihood rather than simply deciding between the binary states of *walking* and *not walking*. If a low likelihood of walking is detected, the algorithm can decide to put more emphasis on the previously detected emotional state instead of trying to calculate a new estimation.

Such an approach is also reasonable in terms of monitoring emotions over somewhat longer periods of time. In this work, only sections of 20 seconds were used, but some applications might require much longer durations. In those situations, the fact that people do not tend to have too high short-term emotional fluctuations has to be taken into account. That means that instead of only analyzing a short window of time like the method currently does, the recent emotional state also has to be taken into account, e.g. through the use of a weighted moving average filter.

Appendices

Appendix A

Technical implementation

A.1 *Emotions* – app for data collection

The functionality of the app is described in section 3.2.3, and an overview of the implementation can be seen in figure A.1.

Android applications are implemented in Java and function through a multitude of components, e.g. *activities*[1] and *services*[2], that work together to achieve desired functionality. While activities normally are used for something the user can do, i.e. they somehow interact with the user, services are normally used for processes happening in the background without requiring the user to interact. *Emotions* uses activities and services for such purposes in the following way:

- **MainActivity** initiates all the functionality of the app. This activity ...
  - starts the service that handles the recording of accelerometer data.
  - gives the user information about what the app is doing.
  - provides access to instructions on how to use the app.
  - receives information from the service when the service has performed a recording.
  - starts the activity that requests the user to provide the emotions he/she is experiencing and receives the result.
  - sends information to the server once a data set is generated.
  - provides the user with a way to shut down the service and exit the app.
- **SensorMonitorService (SMS)** runs as a foreground[3] service and listens to the events from the accelerometer. SMS runs in foreground to ensure that it doesn't get cancelled by the Android operating

[1] https://developer.android.com/guide/components/activities.html
[2] https://developer.android.com/guide/components/services.html
[3] https://developer.android.com/reference/android/app/Service.html#startForeground%28int,%20android.app.Notification%29

system, which happened frequently when the service was run in the background only. See sections 3.2.3 and 3.2.3 for information about the specific functionality.

In order to access the accelerometer, a listener is registered through the use of Android's SensorManager[4], and the values are received by SMS and processed as previously explained.

- **EmotionsActivity** is the interface where the user can provide his or her emotional state when SMS has recorded data. Started by MainActivity, EmotionsActivity records user data and returns to MainActivity, where the data is sent to the server. This is accompanied by a notification to the user, which terminates when the user clicks the send button or after at most one minute has passed.

- Figure A.1 gives an overview of how the app *Emotions* works.

A.2 Server

As explained in section 3.2.3 and Figure A.1, the app sends the data to a server once the user has provided input and confirmed being in motion immediately before. This server is implemented in Java and functions as follows:

- The basis of the server is a Spring MVC project[5], allowing a basic implementation of a web interface where data can be received and transmitted.

- There are two access points for incoming HTTP requests:

  - */send*: Accepts POST requests where a string is received and then stored in the database. For simplicity, the entire string of data that is received from the app is stored as is, wrapped in a class where it is automatically assigned a database id. When the entry is stored in the database, the server also sends the data as an e-mail, providing simultaneously an immediate backup and notification of new data being received.

  - */show*: This access point populates a web site with all the information stored in the database.

- For data storage an implementation of Hibernate[6] is used, with a PostgreSQL[7] database.

- The server is hosted through OpenShift[8], a cloud hosting service provided by Red Hat.

[4]https://developer.android.com/reference/android/hardware/SensorManager.html
[5]http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html
[6]http://hibernate.org/
[7]https://www.postgresql.org/
[8]https://www.openshift.com/

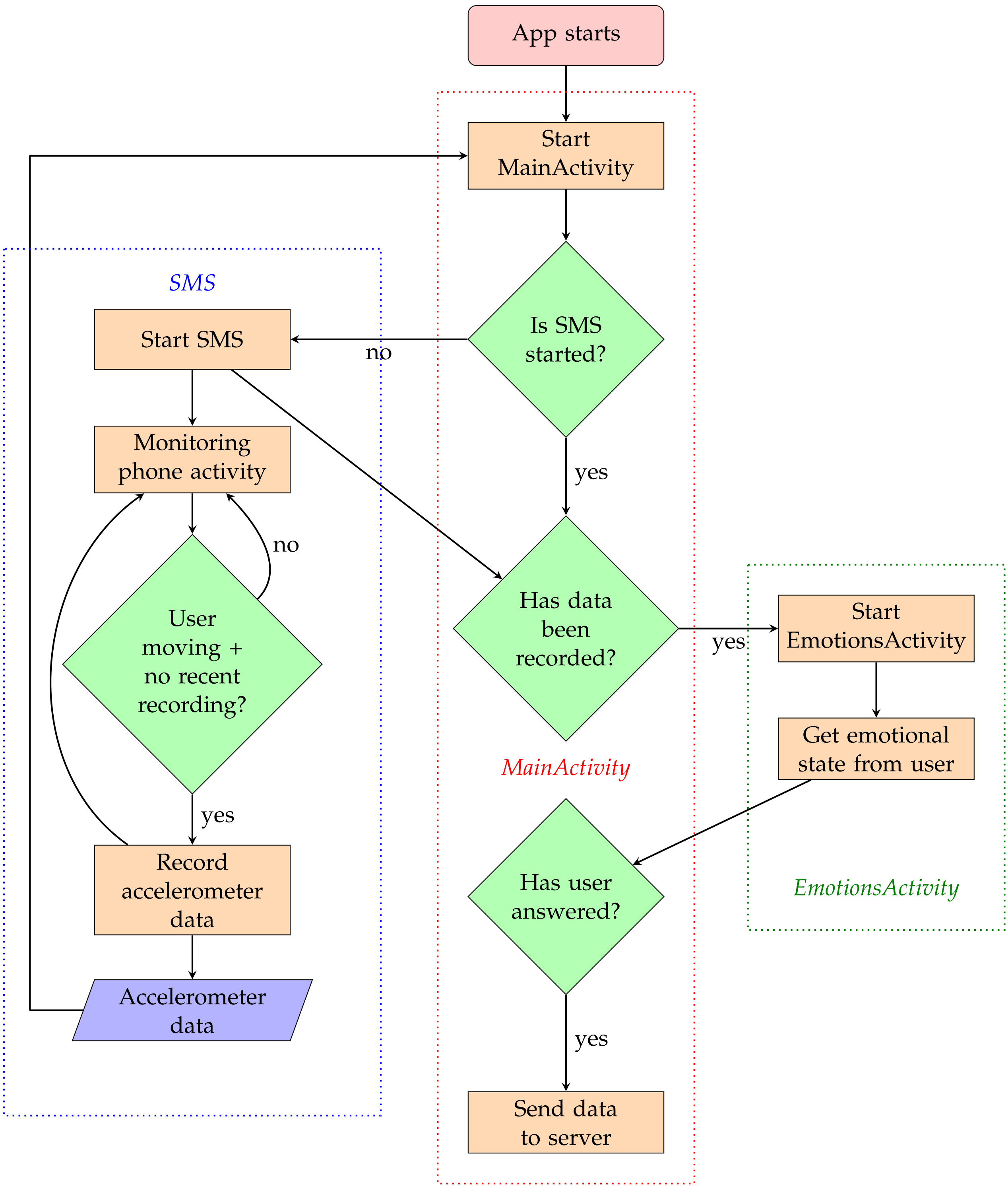


Figure A.1: How the app works

- Figure A.2 gives an overview of how the server works.

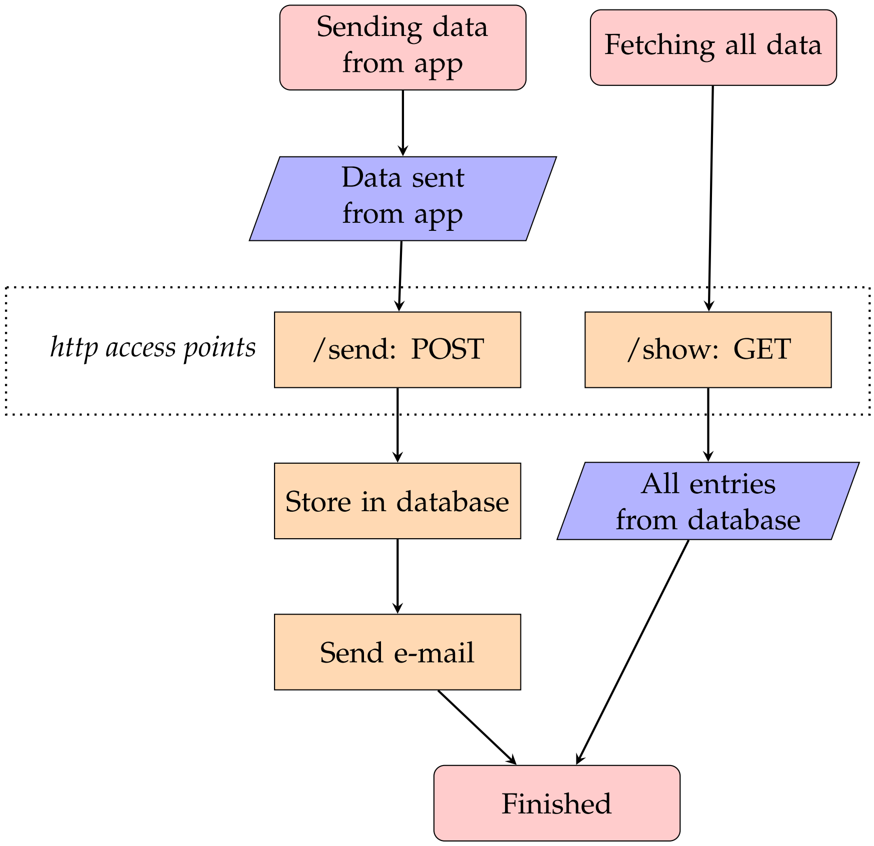


Figure A.2: How the server works

A.3 Classification

For classification, Python was chosen as the programming language. This is due to Python being well suited for scientific and mathematical programming, and personal preferences made this an obvious choice instead of MATLAB, which is another common language used for similar purposes.

The Python library SciPy [63], especially the NumPy [47] module, is used for various array and matrix operations.

Methods for support vector machine and decision tree classification are accessed through a library called scikit-learn [48, 62], an open source machine learning library for Python.

Parameters for the different classifiers were decided experimentally. This was done partly through starting with default options and experimenting manually, and partly through a process called *grid search*. When using grid search, a parameter space is defined for each relevant parameter, in the form of an array containing possible values. Then classification is performed for each possible combination of parameters, and the combination of parameters that results in the highest classification accuracy is reported.

A.3.1 Classification parameters and other implementation details

All code references in this section are written in the Python programming language.

Step segmentation

The following code shows the process of step segmentation:

```
import scipy.signal

step_cadence_limit_factor = 1.5
min_no_consecutive_steps = 4
low_value_threshold = 2.2

def butter(y, critical_frequency):
    b, a = signal.butter(5, critical_frequency, 'low')
    new_y = signal.filtfilt(b, a, y)
    return new_y

def remove_low_values(y, threshold):
    low_indices = y[:] < threshold
    y[low_indices] = 0
    return y

def calculate_maxima(self):
    y = self.butter(self.jerk, 0.05)
    y = self.remove_low_values(copy.deepcopy(y), self.
            low_value_threshold)
    self.maxima_jerk = signal.argrelmax(y)[0]

def find_periods_of_motion(self, maxima, maxima_diff, threshold):
    counter = 0
    result = [[]]
    ongoing = []
    for i in range(len(maxima_diff)):
        if threshold / self.step_cadence_limit_factor < maxima_diff[i
                ] < threshold * self.step_cadence_limit_factor:
            if not maxima[i] in ongoing:
                ongoing.append(maxima[i])
            ongoing.append(maxima[i + 1])
            if len(ongoing) >= self.min_no_consecutive_steps:
                if len(result) > counter:
                    result[counter] = ongoing
                else:
                    result.append(ongoing)
        else:
            ongoing = []
            if counter < len(result) and len(result[counter]) > 0:
                counter += 1
    return result

def calculate_steps_and_periods_of_motion(self):
```

```python
        # Converting time from ms to s
        time = float(self.time) / 1000

        # Normal walking cadence is just above 1 step per second, so we want the distance between steps to be somewhere in that area. values_per_second is the number of accelerometer readings per second, and this will be the expected number of readings per step.
        values_per_second = round(self.size / time)

        maxima = np.asarray(self.maxima_jerk)

        # Find the distances between the maxima
        maxima_diff = [maxima[i + 1] - maxima[i] for i in range(len(maxima) - 1)]

        # Find the sufficiently long consecutive lines of maxima where the distance between any two consecutive maxima is within a certain range of values_per_second.
        periods_of_motion = self.find_periods_of_motion(maxima, maxima_diff, values_per_second)

        x_steps = []
        y_steps = []
        x_periods = []
        y_periods = []
        x_steps_middle = []
        y_steps_middle = []

        if len(periods_of_motion[0]) > 0:
            x = range(len(self.jerk))

            for period in periods_of_motion:
                # Defining the periods of motion
                x_periods.append(x[period[0]:period[-1]])
                y_periods.append(self.jerk[period[0]:period[-1]])

                # Defining each step. This is the simple way of splitting steps between peaks, and the result is not used for feature extraction.
                for i in range(len(period) - 1):
                    x_steps.append(x[period[i]:period[i + 1]])
                    y_steps.append(self.jerk[period[i]:period[i + 1]])

                # Defining each step by splitting in the middle of two peaks. This method is used for later feature extraction.

                # Defining the first step by interpolating:
                higher = int(round((period[0] + period[1]) * 0.5))
                lower = higher - (period[1] - period[0])
                # Ensuring that the interpolation doesnt exceed the limits of the recording
                if lower >= 0:
                    x_steps_middle.append(x[lower:higher])
                    y_steps_middle.append(self.jerk[lower:higher])

                # Defining all the middle steps
                for j in range(len(period) - 2):
```

```
                    lower = int(round((period[j] + period[j + 1]) * 0.5))
                    higher = int(round((period[j + 1] + period[j + 2]) *
                 0.5))
                    x_steps_middle.append(x[lower:higher])
                    y_steps_middle.append(self.jerk[lower:higher])

                # Defining the last step by interpolating:
                lower = int(round((period[-2] + period[-1]) * 0.5))
                higher = lower + (period[-1] - period[-2])
                # Ensuring that the interpolation doesnt exceed the
                   limits of the recording
                if higher <= len(x):
                    x_steps_middle.append(x[lower:higher])
                    y_steps_middle.append(self.jerk[lower:higher])

        self.periods_of_motion = (x_periods, y_periods)
        self.steps = (x_steps, y_steps)
        self.steps_middle = (x_steps_middle, y_steps_middle)

    self.calculate_maxima()
    self.calculate_steps_and_periods_of_motion()
```

Skewness

The inner part of equation 3.13 is calculated using the following implementation:

```
    import scipy.stats.skew
```

Kurtosis

The inner part of equation 3.14 is calculated using the following implementation:

```
    import scipy.stats.kurtosis
```

Power Spectral Density (PSD)

PSD is calculated from the SciPy function *periodogram*, taking the acceleration vector and sampling frequency as input:

```
    f, psd = scipy.signal.periodogram(acceleration,
                 sampling_frequency)
    psd_mean = numpy.mean(psd)
    psd_std = numpy.std(psd)
```

Fourier Transform (FFT)

FFT is calculated from the Numpy function *fft*, taking the acceleration vector as input:

```
fft = numpy.fft.fft(acceleration)
```

Recursive Feature Elimination

```
def perform_feature_selection(data, target):
    from sklearn.feature_selection import RFE
    from sklearn.linear_model import LogisticRegression
    model = LogisticRegression()
    new_num_features = num_features - N # N is the (
            experimentally determined) number of features
            that can be removed without reducing the
            classification accuracy.
    rfe = RFE(model, new_num_features)
    fit = rfe.fit(data, target)
    selected_features = fit.support_ # Returns a boolean array
            indicating which features have been selected
```

Decision tree

The following shows a relevant excerpt of the implementation:

```
from sklearn import tree
clf = tree.DecisionTreeClassifier(min_samples_split=5)
clf.fit(features_train, targets_train)
result = clf.score(features_test, targets_test)
```

This setup was found to work well for the arousal dimension. For the pleasantness dimension, however, the accuracy turned out to be significantly worse than random guess (approximately 35%). That probably has something to do with classification only using one feature for pleasantness, and changing to the following parameter moved the classification results back to the level of random guess:

```
clf = tree.DecisionTreeClassifier(min_samples_split=50)
```

Validation: K-fold cross-validation (K = 5) with $\frac{K-1}{K}$ of samples used for training set and $\frac{1}{K}$ for test set.

K-fold cross-validation divides the data set into k parts. One part is selected as a test set, while the remaining ones are used for training. Once the classifier has trained and produced a result for this division of the data set, another of the k parts is chosen as the test set. This is repeated till all the k parts have been used as a test set, and the final result is the average of all the generated results. For $k = 5$, which is used in this project, it means that each run of the classifier actually has 5 iterations.

Support vector machine

The following shows a relevant excerpt of the implementation:

```
from sklearn import svm
clf = svm.SVC(kernel='poly', degree=4, coef0=4, gamma=0.4,
              decision_function_shape='ovr')
clf.fit(features_train, targets_train)
result = clf.score(features_test, targets_test) * 100
```

Validation: K-fold cross-validation (K = 5) with $\frac{K-1}{K}$ of samples used for training set and $\frac{1}{K}$ for test set.

Multilayer perceptron

The MLP algorithm is using the parameters listed in table A.1.

In this case, the K-fold cross validation also includes a validation set in

Table A.1: MLP parameters

Momentum constant	0.9
Learning rate	0.1
Hidden layers	1
Hidden nodes	10
Validation	K-fold cross-validation (K = 5) with $\frac{K-2}{K}$ of samples used for training set, $\frac{1}{K}$ for validation set and $\frac{1}{K}$ for test set.

addition to training and test sets. Therefore, with $k = 5$ we get $5 * 4 = 20$ combinations of data divisions for each run of the classifier.

Experimenting with different numbers of hidden nodes in the range 4-20 generally seemed to give rather similar results, so the final choice of 10 was considered to be sufficient.

Appendix B

Plots

B.1 Feature plots

The figures in this section are scatter plots of the rescaled feature values calculated from each entry in the data set plotted against the self-reported pleasantness and arousal, respectively.

Acceleration

Figure B.1 shows the acceleration mean and standard deviation.

Top acceleration per step

Figure B.2 shows the top acceleration per step mean and standard deviation.

Jerk

Figure B.3 shows the jerk mean and standard deviation.

Top jerk per step

Figure B.4 shows the top jerk per step mean and standard deviation.

Step duration

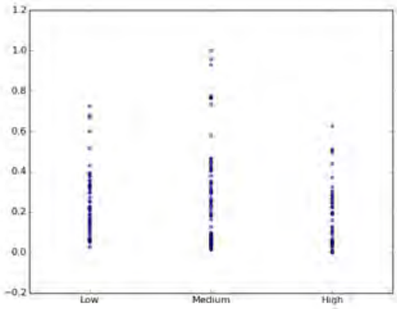
Figure B.5 shows that the step duration mean and standard deviation.

Skewness

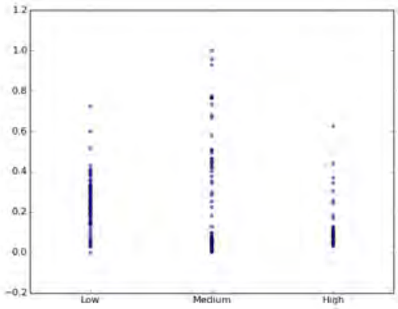
Figure B.6 shows skewness. Extracting these features from the jerk values rather than acceleration values gave similar results.

Kurtosis

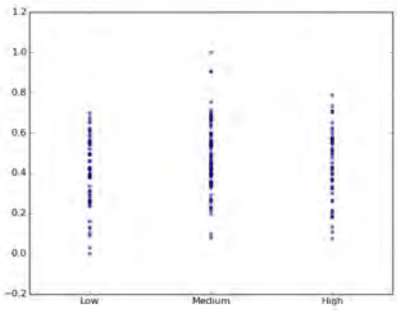
Figure B.7 shows kurtosis. Extracting these features from the jerk values rather than acceleration values gave similar results.



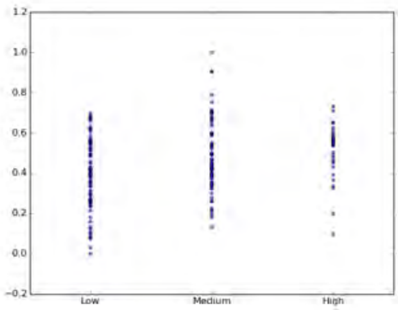
(a) Distribution of acceleration mean as a function of pleasantness.



(b) Distribution of acceleration mean as a function of arousal.

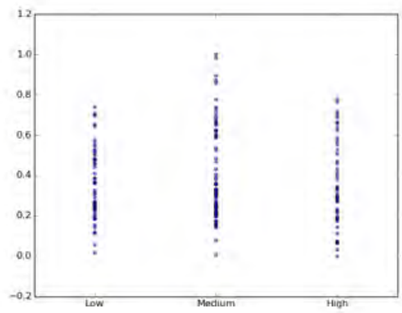


(c) Distribution of acceleration standard deviation as a function of pleasantness.

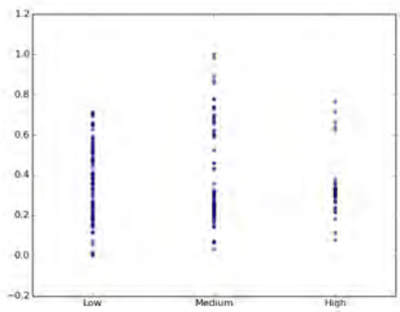


(d) Distribution of acceleration standard deviation as a function of arousal.

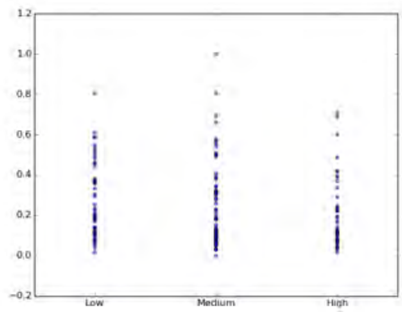
Figure B.1: Acceleration



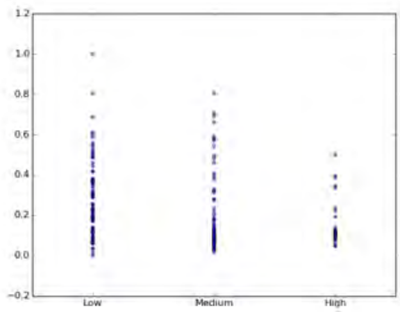
(a) Distribution of top acceleration per step mean as a function of pleasantness.



(b) Distribution of top acceleration per step mean as a function of arousal.

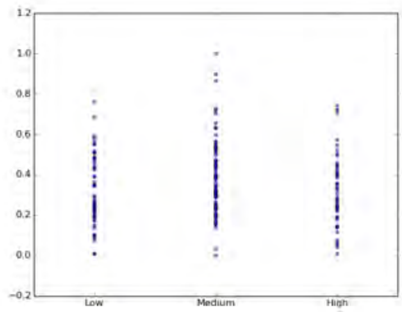


(c) Distribution of top acceleration per step standard deviation as a function of pleasantness.

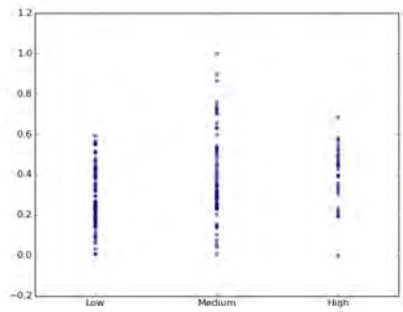


(d) Distribution of top acceleration per step standard deviation as a function of arousal.

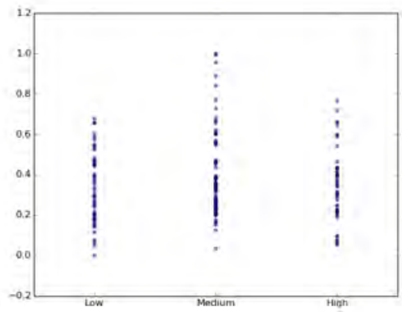
Figure B.2: Top acceleration per step



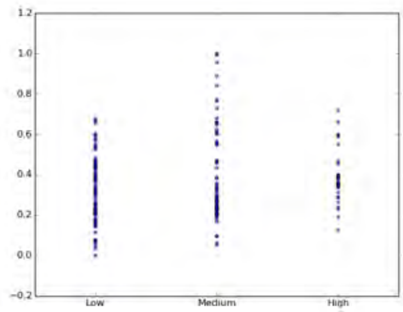
(a) Distribution of jerk mean as a function of pleasantness.



(b) Distribution of jerk mean as a function of arousal.

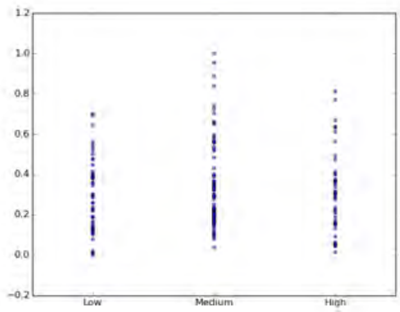


(c) Distribution of jerk standard deviation as a function of pleasantness.

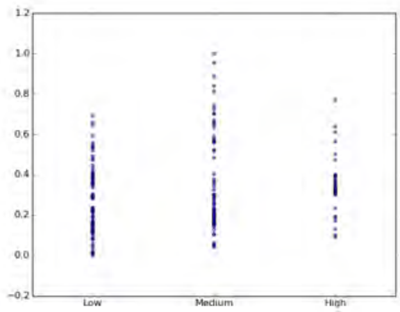


(d) Distribution of jerk standard deviation as a function of arousal.

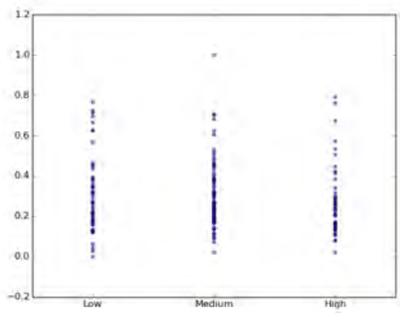
Figure B.3: Jerk



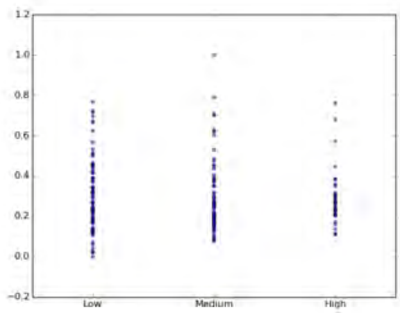
(a) Distribution of top jerk per step mean as a function of pleasantness.



(b) Distribution of top jerk per step mean as a function of arousal.

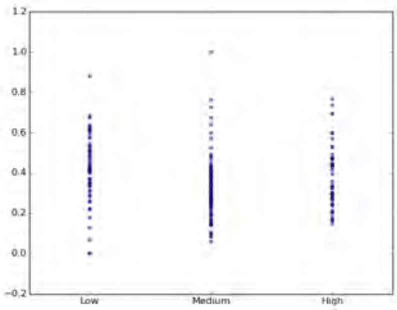


(c) Distribution of top jerk per step standard deviation as a function of pleasantness.

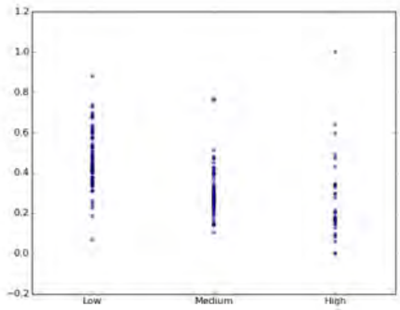


(d) Distribution of top jerk per step standard deviation as a function of arousal.

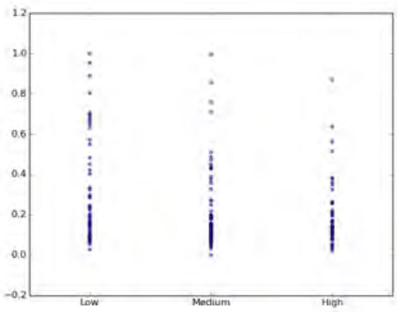
Figure B.4: Top jerk per step



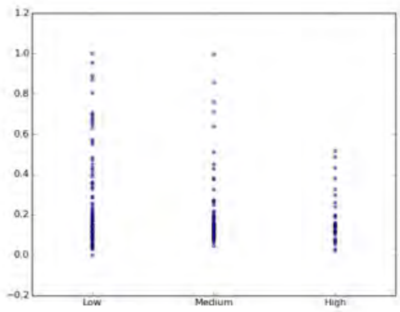
(a) Distribution of step duration mean as a function of pleasantness.



(b) Distribution of step duration mean as a function of arousal.

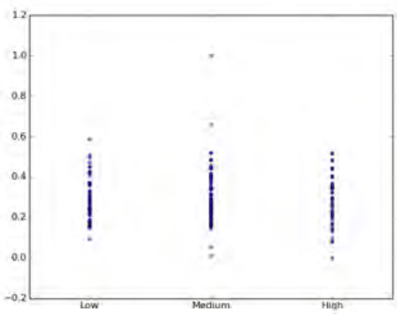


(c) Distribution of step duration standard deviation as a function of pleasantness.

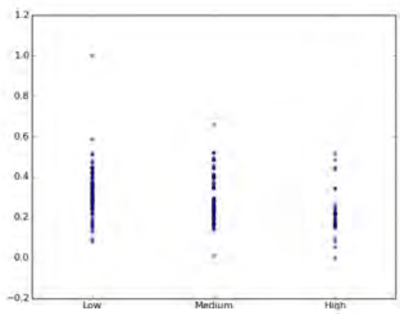


(d) Distribution of step duration standard deviation as a function of arousal.

Figure B.5: Step duration

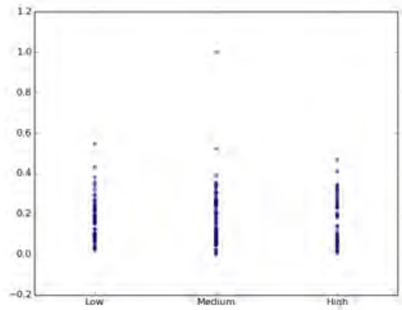


(a) Distribution of skewness as a function of pleasantness.

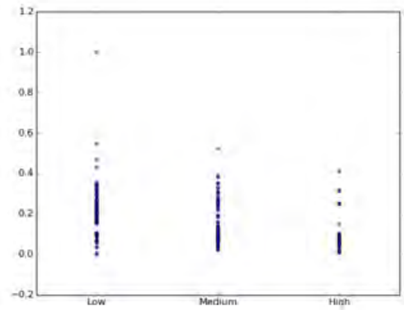


(b) Distribution of skewness as a function of arousal.

Figure B.6: Skewness



(a) Distribution of kurtosis as a function of pleasantness.



(b) Distribution of kurtosis as a function of arousal.

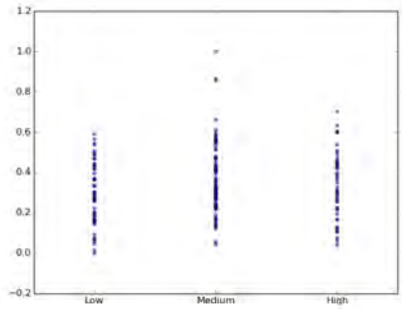
Figure B.7: Kurtosis

Fourier Transform

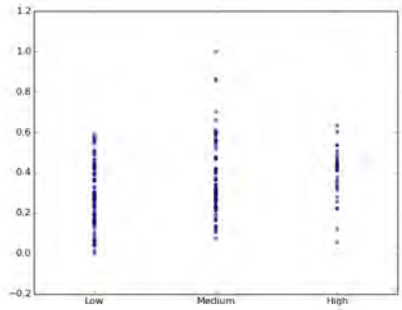
Fourier transform was performed on the acceleration vectors of the recordings. The resulting array of Fourier coefficients is of the same length as the number of data points in a recording, and the first 30 coefficients have been considered. However, they did not seem to have any discriminatory value, so they have not been included as features. Extracting these features from the jerk values rather than acceleration values gave similar results.

Power Spectral Density

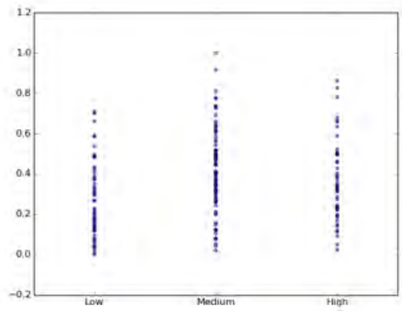
Figure B.8 shows the power spectral density mean and standard deviation. Extracting these features from the jerk values rather than acceleration values gave similar results.



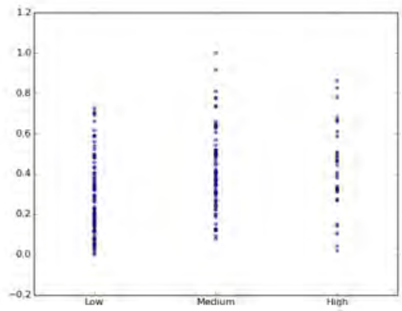
(a) Distribution of power spectral density mean as a function of pleasantness.



(b) Distribution of power spectral density mean as a function of arousal.



(c) Distribution of power spectral density standard deviation as a function of pleasantness.



(d) Distribution of power spectral density standard deviation as a function of arousal.

Figure B.8: Power Spectral Density

Appendix C

Information given to participants

Emotions – datainnsamling til masteroppgave i informatikk

Tusen takk for at du hjelper meg samle inn data til masteroppgaven!

Det eneste du trenger gjøre er å installere appen og – når det passer – svare den når den spør deg om å fylle inn data. Det skal ikke skje så ofte, men forhåpentligvis en gang iblant når du er ute og går og har telefonen i lommen.

Hensikten med prosjektet mitt er å finne ut om telefonen – nærmere bestemt akselerometeret i telefonen – kan brukes til å finne en sammenheng mellom hvordan man føler seg og hvordan man beveger seg når man er ute og går. Resultater fra prosjektet kan blant annet brukes innen **helse og omsorg**, der man kan forsøke å fange opp data fra pasienter som et ledd i å vurdere tiltak.

Ingen personlige data samles inn eller lagres

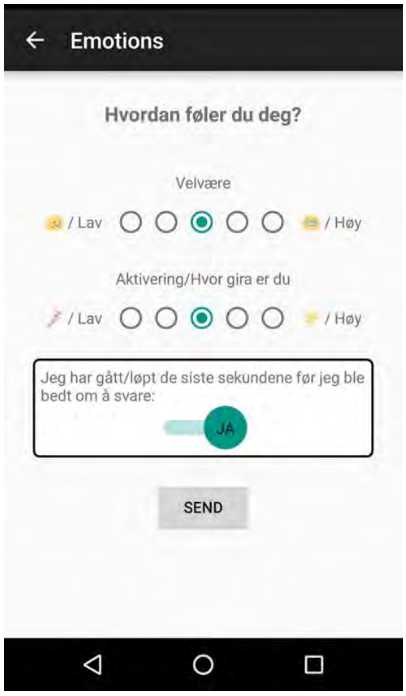
Appen samler inn data fra akselerometeret og det du svarer, men det knyttes ikke på noen måte til deg personlig. Du identifiseres i datasettet kun med en tilfeldig generert id som ikke kan spores tilbake til en enkeltperson.

Hva skal fylles inn

Det er tre ting du skal svare på, der de to første går på hvordan du føler deg, og den siste er der du skal bekrefte at du faktisk har beveget deg rett før målingen.

- **Velvære**: Velvære betyr hvor fornøyd du er. 0 er veldig trist/misfornøyd, 4 er veldig glad/fornøyd.
- **Aktivering**: Aktivering betyr hvor energisk/gira/"på" du føler deg. 0 er veldig avslappet/trøtt/rolig, 4 er veldig energisk/aktivert/gira.
- På siste punkt svarer du ja eller nei på om du nettopp har beveget deg. Til slutt trykker du på "Send".

Et eksempel på en utfylling der personen er i nøytralt humør og nettopp har gått.



Praktisk

For å avslutte appen helt må du trykke på "Avslutt"-knappen. Sveipes den ut til siden vil den fortsatt kjøre i bakgrunnen, og så lenge den kjører kan du se et ikon oppe i hjørnet.

For å sende inn data trenger appen internettilgang. Den bruker veldig lite data, men anbefaler å avslutte appen f.eks. ved utenlandsreiser.

Bibliography

[1] Mohammad Adibuzzaman et al. 'In situ affect detection in mobile devices: a multimodal approach for advertisement using social network'. In: *ACM SIGAPP Applied Computing Review* 13.4 (2013), pp. 67–77.

[2] *Android API: Sensor class*. URL: https://developer.android.com/reference/android/hardware/Sensor.html.

[3] *Android API: Sensors overview*. URL: http://developer.android.com/guide/topics/sensors/sensors_overview.html.

[4] *AndroSensor*. URL: http://www.fivasim.com/androsensor.html.

[5] Daniel Bernhardt and Peter Robinson. 'Detecting emotions from everyday body movements'. In: *Presenccia PhD Sym., Barcelona* (2007).

[6] Rasam Bin Hossain, Mefta Sadat and Hasan Mahmud. 'Recognition of human affection in Smartphone perspective based on accelerometer and user's sitting position'. In: *Computer and Information Technology (ICCIT), 2014 17th International Conference on*. IEEE. 2014, pp. 87–91.

[7] Alberto G Bonomi. 'Physical activity recognition using a wearable accelerometer'. In: *Sensing Emotions*. Springer, 2010, pp. 41–51.

[8] Alberto G Bonomi et al. 'Improving assessment of daily energy expenditure by identifying types of physical activity with a single accelerometer'. In: *Journal of Applied Physiology* 107.3 (2009), pp. 655–661.

[9] Laura L Carstensen and John D Cone. 'Social desirability and the measurement of psychological well-being in elderly persons'. In: *Journal of Gerontology* 38.6 (1983), pp. 713–715.

[10] Ting-Ray Chang, Eija Kaasinen and Kirsikka Kaipainen. 'Persuasive Design in Mobile Applications for Mental Well-Being: Multidisciplinary Expert Review'. In: *Wireless Mobile Communication and Healthcare: Third International Conference, MobiHealth 2012, Paris, France, November 21-23, 2012, Revised Selected Papers*. Ed. by Balwant Godara and Konstantina S. Nikita. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 154–162. ISBN: 978-3-642-37893-5. DOI: 10.1007/978-3-642-37893-5_18. URL: http://dx.doi.org/10.1007/978-3-642-37893-5_18.

[11] Karen Church, Eve Hoggan and Nuria Oliver. 'A study of mobile mood awareness and communication through MobiMood'. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. ACM. 2010, pp. 128–137.

[12] Ernest Cline. *Ready Player One*. Crown Publishers, 2011.

[13] *Comma-separated values*. URL: https://en.wikipedia.org/wiki/Comma-separated_values.

[14] Céline Coutrix and Nadine Mandran. 'Identifying emotions expressed by mobile users through 2D surface and 3D motion gestures'. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM. 2012, pp. 311–320.

[15] Elizabeth Crane and Melissa Gross. 'Motion capture and emotion: Affect detection in whole body movement'. In: *International Conference on Affective Computing and Intelligent Interaction*. Springer. 2007, pp. 95–101.

[16] Liqing Cui et al. 'Identifying Emotion from Natural Walking'. In: *arXiv preprint arXiv:1508.00413* (2015).

[17] Robert F. Dickerson, Eugenia I. Gorlin and John A. Stankovic. 'Empath: A Continuous Remote Emotional Health Monitoring System for Depressive Illness'. In: *Proceedings of the 2Nd Conference on Wireless Health*. WH '11. San Diego, California: ACM, 2011, 5:1–5:10. ISBN: 978-1-4503-0982-0. DOI: 10.1145/2077546.2077552. URL: http://doi.acm.org/10.1145/2077546.2077552.

[18] Ed Diener et al. 'Response artifacts in the measurement of subjective well-being'. In: *Social Indicators Research* 24.1 (1991), pp. 35–56.

[19] Paul Ekman. 'An argument for basic emotions'. In: *Cognition & emotion* 6.3-4 (1992), pp. 169–200.

[20] Paul Ekman. *Handbook of cognition and emotion*. 1999. Chap. 3, pp. 45–60.

[21] *Fast Fourier Transform*. URL: https://en.wikipedia.org/wiki/Fast_Fourier_transform.

[22] *Fourier Transform*. URL: https://en.wikipedia.org/wiki/Fourier_transform.

[23] Liyu Gong et al. 'Recognizing Affect from Non-stylized Body Motion Using Shape of Gaussian Descriptors'. In: *Proceedings of the 2010 ACM Symposium on Applied Computing*. SAC '10. Sierre, Switzerland: ACM, 2010, pp. 1203–1206. ISBN: 978-1-60558-639-7. DOI: 10.1145/1774088.1774341. URL: http://doi.acm.org/10.1145/1774088.1774341.

[24] *Google image search on Ekman's six basic emotions*. URL: https://en.wikipedia.org/wiki/File:Angry_face.png, https://commons.wikimedia.org/wiki/File:Tox_inedible.svg, https://en.wikipedia.org/wiki/Smiley#/media/File:Smiley.svg, https://pixabay.com/en/face-eyes-caricature-comic-fear-8685/, https://commons.wikimedia.org/wiki/File:Sadness.png, https://commons.wikimedia.org/wiki/File:Gnome3-surprise.svg.

[25] Illapha Cuba Gyllensten and Alberto G Bonomi. 'Identifying types of physical activity with a single accelerometer: evaluating laboratory-trained algorithms in daily life'. In: *Biomedical Engineering, IEEE Transactions on* 58.9 (2011), pp. 2656–2663.

[26] Chia-Ming Hsu, Ting Ting Chen and Jia Sheng Heh. 'Emotional and Conditional Model for Pet Robot based on Neural Network'. In: *Ubi-Media Computing and Workshops (UMEDIA), 2014 7th International Conference on*. IEEE. 2014, pp. 305–308.

[27] Carroll E Izard. *Human emotions*. Plenum, 1977. ISBN: 978-1-4899-2211-3.

[28] Joris H. Janssen, Egon L. van den Broek and Joyce H. D. M. Westerink. 'Tune in to your emotions: a robust personalized affective music player'. In: *User Modeling and User-Adapted Interaction* 22.3 (2012), pp. 255–279. ISSN: 1573-1391. DOI: 10.1007/s11257-011-9107-7. URL: http://dx.doi.org/10.1007/s11257-011-9107-7.

[29] Natasha Jaques et al. 'Multi-task, Multi-Kernel Learning for Estimating Individual Wellbeing'. In: *Proc. NIPS Workshop on Multimodal Machine Learning, Montreal, Quebec*. 2015.

[30] Natasha Jaques et al. 'Predicting students' happiness from physiology, phone, mobility, and behavioral data'. In: *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*. IEEE. 2015, pp. 222–228.

[31] Patrik N Juslin and John A Sloboda. *Music and emotion: Theory and research.* Oxford University Press, 2001.

[32] Eiman Kanjo, Luluah Al-Husain and Alan Chamberlain. 'Emotions in context: examining pervasive affective sensing systems, applications, and analyses'. In: *Personal and Ubiquitous Computing* 19.7 (2015), pp. 1197–1212. ISSN: 1617-4917. DOI: 10.1007/s00779-015-0842-3. URL: http://dx.doi.org/10.1007/s00779-015-0842-3.

[33] Asha Kapur et al. 'Affective Computing and Intelligent Interaction: First International Conference, ACII 2005, Beijing, China, October 22-24, 2005. Proceedings'. In: ed. by Jianhua Tao, Tieniu Tan and Rosalind W. Picard. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. Chap. Gesture-Based Affective Computing on Motion Capture Data, pp. 1–7. ISBN: 978-3-540-32273-3. DOI: 10.1007/11573548_1. URL: http://dx.doi.org/10.1007/11573548_1.

[34] Michelle Karg et al. 'A Two-fold PCA-Approach for Inter-Individual Recognition of Emotions in Natural Walking.' In: *MLDM Posters*. 2009, pp. 51–61.

[35] Vlada Kugurakova et al. 'Anthropomorphic Artificial Social Agent with Simulated Emotions and its Implementation'. In: *Procedia Computer Science* 71 (2015), pp. 112–118.

[36] *Kurtosis*. URL: https://en.wikipedia.org/wiki/Kurtosis.

[37] R.J. Larsen and D.M. Buss. *Personality Psychology: Domains of Knowledge about Human Nature*. 5th ed. McGraw-Hill, 2014. ISBN: 9780078035357.

[38] Robert LiKamWa et al. 'Moodscope: building a mood sensor from smartphone usage patterns'. In: *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM. 2013, pp. 389–402.

[39] Hugo Lövheim. 'A new three-dimensional model for emotions and monoamine neurotransmitters'. In: *Medical hypotheses* 78.2 (2012), pp. 341–348.

[40] Yingliang Ma, Helena M Paterson and Frank E Pollick. 'A motion capture library for the study of identity, gender, and emotion perception from biological motion'. In: *Behavior research methods* 38.1 (2006), pp. 134–141.

[41] Yuanchao Ma et al. 'Daily mood assessment based on mobile phone sensing'. In: *Wearable and implantable body sensor networks (BSN), 2012 ninth international conference on*. IEEE. 2012, pp. 142–147.

[42] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC press, 2015.

[43] Johannes Michalak et al. 'Embodiment of sadness and depression—gait patterns associated with dysphoric mood'. In: *Psychosomatic medicine* 71.5 (2009), pp. 580–587.

[44] Joann M Montepare, Sabra B Goldstein and Annmarie Clausen. 'The identification of emotions from gait information'. In: *Journal of Nonverbal Behavior* 11.1 (1987), pp. 33–42.

[45] *MPU-6500*. URL: http://www.invensense.com/products/motion-tracking/6-axis/mpu-6500/.

[46] *MPU-6500-Datasheet2*. URL: http://43zrtwysvxb2gf29r5o0athu.wpengine.netdna-cdn.com/wp-content/uploads/2015/02/MPU-6500-Datasheet2.pdf.

[47] *Numpy*. URL: http://www.numpy.org/.

[48] F. Pedregosa et al. 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[49] Rosalind W Picard. *Affective computing*. MIT press Cambridge, 1997.

[50] *Pokémon Go*. URL: http://www.pokemongo.com/.

[51] *Power Spectral Density*. URL: https://en.wikipedia.org/wiki/Spectral_density#Power_spectral_density.

[52] *PowerManager.PARTIAL_WAKE_LOCK*. URL: http://developer.android.com/reference/android/os/PowerManager.html#PARTIAL_WAKE_LOCK.

[53] Stephen J Preece et al. 'Activity identification using body-mounted sensors—a review of classification techniques'. In: *Physiological measurement* 30.4 (2009), R1.

[54] J. Ross Quinlan. 'Induction of decision trees'. In: *Machine learning* 1.1 (1986), pp. 81–106.

[55] Kiran K Rachuri et al. 'EmotionSense: a mobile phones based adaptive platform for experimental social psychology research'. In: *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM. 2010, pp. 281–290.

[56] *Recursive Feature Elimination*. URL: http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html.

[57] Michael D Robinson and Lisa Feldman Barrett. 'Belief and feeling in self-reports of emotion: Evidence for semantic infusion based on self-esteem'. In: *Self and Identity* 9.1 (2010), pp. 87–111.

[58] Catherine A Roster, Lorenzo Lucianetti and Gerald Albaum. 'Exploring Slider vs. Categorical Response Formats in Web-Based Surveys'. In: *Journal of Research Practice* 11.1 (2015), p. 1.

[59] James A Russell. 'A circumplex model of affect.' In: *Journal of personality and social psychology* 39.6 (1980), pp. 1161–1178.

[60] James A Russell. 'Evidence of convergent validity on the dimensions of affect.' In: *Journal of personality and social psychology* 36.10 (1978), p. 1152.

[61] MM Samson et al. 'Differences in gait parameters at a preferred walking speed in healthy subjects due to age, height and body weight'. In: *Aging Clinical and Experimental Research* 13.1 (2001), pp. 16–21.

[62] *scikit-learn*. URL: http://scikit-learn.org/stable/index.html.

[63] *scipy*. URL: https://www.scipy.org/.

[64] Claude E Shannon. 'Communication in the presence of noise'. In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21.

[65] Takuma Shibata and Yoshifumi Kijima. 'Emotion recognition modeling of sitting postures by using pressure sensors and accelerometers'. In: *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE. 2012, pp. 1124–1127.

[66] *Siri*. URL: http://www.apple.com/ios/siri/.

[67] *Skewness*. URL: https://en.wikipedia.org/wiki/Skewness.

[68] Jie Tang et al. 'Quantitative study of individual emotional states in social networks'. In: *Affective Computing, IEEE Transactions on* 3.2 (2012), pp. 132–144.

[69] David L Thomas and Ed Diener. 'Memory accuracy in the recall of emotions.' In: *Journal of Personality and Social Psychology* 59.2 (1990), p. 291.

[70] Silvan S Tomkins. 'Affect theory'. In: *Approaches to emotion* 163 (1984), pp. 163–195.

[71] Alan M Turing. 'Computing machinery and intelligence'. In: *Mind* 59.236 (1950), pp. 433–460.

[72] *WakeLock*. URL: http://developer.android.com/training/scheduling/wakelock.html.

[73] David Watson. *Mood and temperament*. Guilford Press, 2000.

[74] David Watson, Lee A Clark and Auke Tellegen. 'Development and validation of brief measures of positive and negative affect: the PANAS scales.' In: *Journal of personality and social psychology* 54.6 (1988), p. 1063.

[75] David Watson and Auke Tellegen. 'Toward a consensual structure of mood.' In: *Psychological bulletin* 98.2 (1985), p. 219.

[76] Shumei Zhang et al. 'Activity monitoring using a smart phone's accelerometer with hierarchical classification'. In: *Intelligent Environments (IE), 2010 Sixth International Conference on*. IEEE. 2010, pp. 158–163.

[77] Zhan Zhang et al. 'Emotion recognition based on customized smart bracelet with built-in accelerometer'. In: *PeerJ PrePrints* 4 (2016), e1650v1.

[78] Philippe Zimmermann et al. 'Affective computing—a rationale for measuring mood with mouse and keyboard'. In: *International journal of occupational safety and ergonomics* 9.4 (2003), pp. 539–551.