Classification of Respiratory Data:

Classifying and analyzing respiratory data to present as feedback towards cultivating habitual diaphragmatic breathing

Student: *Arnav Mundkur*
Student number: *s1552236*
Study: *Bachelor Creative Technology*

Supervisor: *dr. ir. Erik Faber*
Critical Observer: *Ainara Martinez Garde*
Client: *Ben Bulsink*

5th July 2018
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
University of Twente

Abstract

Diaphragmatic breathing is a special style of breathing and an aspect central to eastern practices such as Yoga, Tai-Chi and Qigong. Diaphragmatic breathing incorporates the abdomen during inhalation, causing it to expand instead of the chest. Mr. Bulsink, an entrepreneur and inventor based in Enschede, is enthusiastic about diaphragmatic breathing and developed a device, the Airleviate, to measure a wearer's breathing using Respiratory Inductance Plethysmography (RIP). His request was to have data recorded with the Airleviate to be analyzed so that features regarding breathing could be presented to the wearer as feedback. This led to the research question: *"How can Respiratory Inductance Plethysmography data be classified and analyzed to be used as feedback towards cultivating habitual diaphragmatic breathing?"*

Background research showed that diaphragmatic breathing is used as a medical intervention in treating conditions such as oxidative stress and anxiety. The type of classification model used to classify respiration depended on the goal of the study and the features the classifier was trained on. Literature showed that Respiratory Inductance Plethysmography has been validated by the scientific community as a method to measure respiration. Through interviews and brainstorms, it was found that there were specific breathing features of interest, such as the times of a diaphragmatic breathing moment and a quality factor, representing a user's diaphragmatic breathing performance during the day.

A program was developed to process, classify and analyze breathing data by segmenting the data by minute using a sliding window of one minute. Breathing features were extracted from the data and written to a file to be sent to the user interface, another Creative Technology graduation project being done by Florian Naumilkat. During evaluation, the program passed all the code tests it was put through. The methodology to process the data and the approach to classify diaphragmatic breathing were validated by two experts in the fields of data analysis and research on breathing. The program was put through a "real world" trial, where it was found that the program was able to identify some planned diaphragmatic breathing moments. However, during the recording, errors were introduced due to slippage of the chest band, resulting in classification errors. The main requirements of classifying breathing as diaphragmatic, repetitive and the motion of the user, and exporting breathing features to a text file for the interface were all met.

In conclusion, this report introduces a method to classify diaphragmatic breathing using a Support Vector Machine. The breathing signal is cleaned before classification, using a Bandpass filter on the ranges 0.2 - 0.7 Hz. A periodogram is used to extract the main breathing frequency of the signal per minute to evaluate it on repetitiveness. Features such as a quality factor, and the times of diaphragmatic breathing moments are of interest of users for feedback and are extracted.

For future work, the validity of the program should be further tested by fastening the RIP bands and recording longer breathing sessions with more planned diaphragmatic breathing moments. Knowledge transfer between models should be investigated to see if one user's breathing can be used to predict

another’s. The features regarding movement and repetitive breathing should be extracted from the data to provide the wearer with a more holistic view of their breathing in relation to their activities. A different approach to analyzing breathing by looking at the individual breaths and characterizing it using other features could be investigated and its accuracy compared to the approach taken in this project.

Acknowledgements

There are a few individuals I would like to thank for their participation and interest in this project. I would like to begin by thanking my supervisor, Erik Faber, for his constant support, guidance and enthusiasm in the project. Next I would like to thank my client, Mr. Bulsink, for his interest in the progress with the development of the program, technical support with the Airleviate and input for the features of interest for classification. I would like to thank fellow Creative Technology student Florian Naumilkat for his input, energy and inspiration.

I would especially like to thank Mrs. Garde for providing guidance for the classification of breathing, sharing her expertise and experience in the field, and her feedback on my approach to classifying diaphragmatic breathing. I would like to thank Mr. Kamilaris for his expertise in data processing and reviewing my methodology.

Furthermore, I would like to thank Mrs. Ineke Ter Hedde for providing me with a thorough understanding of diaphragmatic breathing and sharing her experience with it. I would like to thank Mr. Parviz Sassanian for providing me with the more holistic and almost spiritual effects of diaphragmatic breathing, that could not be found in literature.

Lastly, I would like to thank the participants in this study. They took time out of their schedules to participate in a lengthy test, and provided insightful details about the Airleviate bands that I hadn’t experienced.

Table of contents

Abstract 2

Acknowledgements 4

Table of contents 6

1. Introduction 11

1.1 Problem description 11

1.2 Context Analysis 12

1.3 Research Questions 14

1.4 Report structure 15

2. Background Research 17

2.1 Literature Research 17

2.2 Expert Interviews 31

2.3 Market Research 33

2.4 Decisions for Ideation 40

3. Methodology and Techniques 41

3.1 Creative Technology Design Process 41

3.2 Brainstorms 43

3.3 Expert Review 44

3.4 Interviews 44

3.5 Stakeholder Analysis 44

3.6 iPACT Analysis 45

3.7 Requirements Analysis 46

3.8 FICS Analysis 46

3.9 System Architecture 47

3.10 Cognitive Walkthrough 47

3.11 Activity Diagram 47

3.12 Evaluation 47

4. Ideation 49

4.1 Stakeholder Analysis 49

4.2 Individual Brainstorm 52

4.3 Group brainstorm 53

4.4 Stakeholder Interviews 54

4.5 Expert Review 57

4.6 Preliminary Concept 58

4.7 iPACT Analysis 59
4.8 Preliminary Requirements 63
4.9 Conclusion 65

5. Specification 67
5.1 FICS analysis 67
5.2 Cognitive Walkthrough 68
5.3 Activity Diagram 70
5.4 System Architecture 72
5.5 Terminology and analysis decisions 78
5.6 Requirements analysis: MoSCoW and Functional - Non-Functional 80

6. Realization 83
6.1 Development 83
6.2 Classifier accuracy tests 85
6.3 Program decomposition 87

7. Evaluation 91
7.1 Code Testing 91
7.2 Expert Validation 99
7.3 Program Verification Test 101
7.4 Functional Test 106
7.5 General Discussion 108
7.6 Third Requirements Iteration 111
7.7 Evaluation Conclusion 112

8. Conclusion 113
8.1 Conclusion 113

9. Recommendations for Future Work 117
9.1 Recommendations 117

10. References 120

11. Appendix 128
11 A. Expert Interview Questions 128
11 B. Individual Brainstorm 131
11 C. Stakeholder Interview Questions 132
11 D. Classifier Accuracy Tests 134
11 E. Informed Consent Form 137
11 F. Program code 140

Page intentionally left blank.

Page intentionally left blank.

1. Introduction

The first section of this report will introduce the problem and current situation of diaphragmatic breathing. This is followed by a short context analysis where terminology and concepts used frequently in the report are be explained. Following from the terminology, the research questions that this project aims to answer are listed followed by a description of the structure of the rest of the report.

1.1 Problem description

Life begins with the first breath and ends with the last. Breathing is a fundamental process that, consciously or unconsciously, humans perform every day and its pattern and intensity changes based on the activities being done. Over the centuries, various breathing techniques were developed as parts of meditation, yoga and martial arts in regions that form modern day India and China. Practices such as Tai-Chi, Sudarshan Kriya Yoga, Pranayama Yoga and Qigong exercises utilize various breathing techniques in combination with stretches or movements that have applications in health care (Li and Yeh 2005; Tolahunase, Sagar and Dada 2017).

Breathing, though a natural process performed subconsciously, can be performed suboptimally. “Dysfunctional breathing” is an umbrella term for changes in breathing patterns resulting in temporary or chronic symptoms that may or may not be respiratory in nature (Depiazzi and Everard, 2016; Jones et al., 2013). Indicators of dysfunctional breathing are predominantly upper-thoracic (upper-chest) breathing, frequent or deep sighs, mouth breathing and use of supporting muscles (Chaitow et al., 2002 as cited by Courtney, van Dixhoorn and Cohen, 2008). Breathing plays an important role in spinal stabilization and posture and dysfunctional breathing has been shown to contribute to pain and motor control deficits (Bradley and Esformes 2014). Dysfunctional breathing is common in individuals suffering from asthma and asthmatic symptoms and research has been done into breathing retraining for asthmatic patients incorporating diaphragmatic, nasal and slow breathing (Bruton et al., 2018; Arden-Close et al., 2017; Courtney 2017).

Ben Bulsink, an entrepreneur and product developer based in Enschede with a passion for healthy living, approached the University of Twente with the project of recording and classifying breathing patterns for a training service to teach users to employ diaphragmatic breathing on a daily basis. He wishes to cultivate the habit of breathing diaphragmatically more regularly as opposed to solely employing “upper-chest breathing” or thoracic breathing. The measurement device he designed to record inhalation and exhalation data of the abdominal and thoracic regions, a medical measurement process known as Respiratory Inductance Plethysmography, will be used for data collection.

Anyone wishing to experience the physiological benefits of diaphragmatic breathing can benefit from this training. This device can be used to train diaphragmatic breathing in a number of contexts ranging from everyday activities to sports and relaxation. It can also be used in medical settings to train diaphragmatic breathing for studies into its effects on various conditions such as oxidative stress, anxiety and blood pressure and be used in correcting dysfunctional breathing.

1.2 Context Analysis

The topic of this research report is classifying respiratory data. The data is acquired, as mentioned above, using a method called Respiratory Inductance Plethysmography. This section will briefly introduce the concept of Respiratory Inductance Plethysmography, as well as introduce the concept of classification and provide a few examples of the various types of classification.

1.2.1 What is Respiratory Inductance Plethysmography?

Respiratory Inductance Plethysmography, hereon referred to as RIP, is a method used to measure respiratory tidal volume and other respiratory measurements in humans and animals (Cohen et al., 1997; Murphy, Renninger and Schramek, 2010). Cohen et al., (1997) explain that RIP measures the changes in thoracic (upper-chest) and abdominal cross-sectional areas to indirectly measure tidal volume or ventilation. The self-inductance of the elastic belts or bands containing insulated wires is measured to approximate cross-sectional area. A calibration scheme is then used to form a linear combination of RIP(thoracic) and RIP(abdominal) (Chadha et al., 1982 as cited by Cohen et al., 1997) based on the assumption that the abdomen and rib cage constitute two mechanical degrees of freedom in the ventilation system. The setup is shown in the image below:

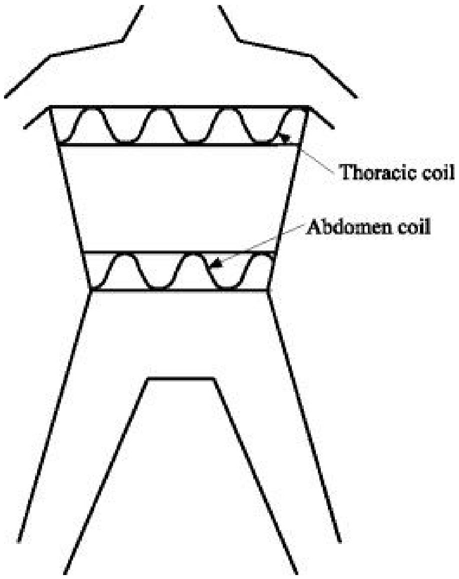


Figure 1.1: *RIP bands positioning for tidal volume measurement (Sensors for Wearable Systems, 2010)*

1.2.2 What is classification?

Classification falls under the category of Machine Learning, and is the concept of a program being fed a set of values or parameters which it then processes to “classify” or label the object whose parameters were given, into known categories. Algorithms are “trained” with data where the parameters are accompanied by the label to which that object belongs. This is called Supervised Learning and thus classification falls under the category of Supervised Learning algorithms. In contrast, Unsupervised Learning is when algorithms are not trained on any classified or labelled data and are used to find hidden structures in data by producing a relationship or trend between the data points (Lopez de Mantaras and Armengol, 1998). Classification is done in two phases. In the first phase, a classification algorithm is fed a training data set. The resulting model is then, in the second phase, tested against a labeled data set in order to test its accuracy and performance.

Classification is necessary for the purpose of this research, as an algorithm will need to be trained on data collected during the diaphragmatic breathing training sessions. After the professional has taught the technique to the participant and confirmed that they are performing it properly, RIP data will be collected to set a personal goal for the participant to strive towards. After being trained on the data, the algorithm will be able to distinguish between diaphragmatic breathing and thoracic (chest) breathing. There are several roles a human plays in the classification process. The researcher must find appropriate data to train the classifier on, and ensure that it is labelled correctly. The data should also cover all the possible cases in order to assure classification accuracy in every scenario. For the scope of this project, a supervised learning algorithm will be used for classification.

1.3 Research Questions

After discussing the goals and scope of this project with Mr. Bulsink, several research questions were formulated around a central question:

“How can Respiratory Inductance Plethysmography data be classified and analyzed to be used as feedback towards cultivating the habit of diaphragmatic breathing?”

With sub-questions addressing the topics of the health benefits of diaphragmatic breathing, measurement of breathing using RIP, classification of RIP data and habit forming for background information and sub-questions regarding the implementation of a solution:

“What are the physiological health benefits of diaphragmatic breathing?”

“Why is RIP used for recording breathing data?”

“How can respiratory data be classified?”

“How is a habit formed?”

“*What kinds of breathing features would users like to be presented for the purpose of feedback?*”

“How can the validity of the program be verified?”

The first sub-question will provide knowledge of the short and long term physiological health benefits of diaphragmatic breathing. The second sub-question will help provide some background on the use of RIP for measuring breathing including shortcomings of the method as well as sources for errors. Once the data has been collected, the next question concerns the classification of the data whether there is any signal preprocessing done before classification. Automaticity or habit formation will be looked into, to understand what needs to be done in order to cultivate a new habit such as breathing diaphragmatically as part of normal breathing routines.

Users will be involved for their opinion on the most useful breathing features that they would like to receive for feedback to answer the next research question. Finally, the validity of the classification program written will be commented on by testing the code, approaching experts for their opinion on the solution and testing the program in a “real world” situation.

1.4 Report structure

This report will begin with an introduction on diaphragmatic breathing and Respiratory Inductance Plethysmography and then move to Background Research, where a literature review will be conducted to answer the various sub-questions posed above. This is followed by the Methods and Techniques section, describing design methodologies and requirements analysis frameworks used in the ideation and specification phases. Next is the Ideation phase, where ideas are generated regarding the algorithm, feature extraction and program testing protocol. The requirements drawn up in the Ideation phase are then evaluated in the Specification phase after which the prototype is realized in the Realization phase and testing is carried out in the Evaluation phase. The Evaluation phase also consists of validation form experts in the fields of data analysis and research on breathing, as well as a “real world” test that the program is put through. The results of the evaluation are then presented and the report will end with the Conclusion followed by Recommendations for further work.

2. Background Research

This report will begin by covering some of the benefits of diaphragmatic breathing, which employs the diaphragm/abdomen during inhalation and exhalation more than the chest. Abdominal breathing is not identical to diaphragmatic breathing and actually falls under the category of the latter, but will be used interchangeably in the scope of this report. A literature review will be conducted to understand the various contexts where diaphragmatic breathing has been used as an intervention and will look at physiological benefits of its practice.

2.1 Literature Research

To guide this review, a number of research questions need to be answered pertaining to classification, measurement, diaphragmatic breathing and habit formation. In order for the service of diaphragmatic breathing retraining to be deemed necessary, the health benefits of diaphragmatic breathing should be known. This section will answer the question:
"*What are the physiological health benefits of diaphragmatic breathing?*"

This section will begin by discussing literature regarding diaphragmatic breathing and its physiological effects. The main question of this section revolves around how people can be convinced to adopt diaphragmatic breathing into their daily routines. The literature is analyzed from the viewpoints of the conditions diaphragmatic breathing is used to prevent and treat, the duration of the intervention and its physiological effects. While conducting a search into the literature, it was found that the topics of the studies were focused on very specific contexts. Therefore, the research will cover three contexts; one of which is Asthma and Asthmatic Symptoms, which relates directly to dysfunctional breathing. The other two contexts affect the greater population namely Oxidative Stress, and Anxiety and Blood Pressure.
The research will then look into how RIP has been used to measure breathing in literature. The sub-questions used to answer this concern the positioning of the RIP bands, the causes for error during measurement, what the RIP was compared against and what the conclusions of the study were.
After that, literature on the classification in the field of breathing and respiratory motion will be investigated. Questions used to guide the review relate to the type of classification used, the features extracted from the data and the type of signal preprocessing done.
Finally, literature on habit formation will be covered to understand how best to cultivate the habit of diaphragmatic breathing. Three aspects of the literature were investigated, namely how a habit is formed, how long it takes to form a habit and the type of feedback used in teaching diaphragmatic breathing.

2.1.1 Physiological effects of Diaphragmatic Breathing

2.1.1.1 Oxidative Stress

The explanations of the effects of diaphragmatic breathing on oxidative stress vary significantly between studies. The Merriam Webster-online medical dictionary defines Oxidative stress as “physiological stress on the body that is caused by the cumulative damage done by free radicals inadequately neutralized by antioxidants and that is held to be associated with aging” (Merriam-webster.com, 2018). Oxidative stress is triggered by activities like eating a meal or exercising.

Martarelli, Cocchioni, Scuri and Pompei (2011a) investigated oxidative stress in a postprandial context (after eating a meal) and state in their conclusion: “diaphragmatic breathing, likely through the activation of the Parasympathetic Nervous System (PNS) increases insulin, reduces glycemia and ROS production”. This conclusion does not agree with a similar study by Martarelli, Cocchioni, Scuri and Pompei (2011b) who state in their conclusion: “the mechanism by which relaxation might induce an increase in melatonin levels is uncertain” and “different mechanisms could be involved”. They claim that the increase in melatonin levels (a powerful antioxidant), can be attributed to the reduced cortisol levels they measured. On the other hand, Hegde et al., (2012) argue that the incorporated principle of “pranayama” (regulation of breathing in yogic practices) with relaxation was the cause of the clinically significant reduction Malondialdehyde, their marker for oxidative stress. From the literature discussed above, in the context of oxidative stress, there is no concrete explanation for the effects of breathing on reduced ROS activity.

There is no agreed amount of time that diaphragmatic breathing needs to be done in order to avoid the effects of oxidative stress. Martarelli et al., (2011a) trained their participants a few days before the experiments and had them perform diaphragmatic breathing in a quiet place for 40 minutes during the experiment. Martarelli et al., (2011b) instructed their participants to perform diaphragmatic breathing for 1 hour after their workout. On the other hand, Hegde et al., (2012) instructed their participants to practice diaphragmatic breathing for 15-20 minutes a day for three months. The duration of the study by Hegde et al., (2012) was much longer than the other two and studied the long term effects of diaphragmatic breathing. However what was unclear from their study was for how long the benefits lasted after diaphragmatic breathing stopped. In the case of Martarelli et al., (2011a); two hours after the meal, the difference in glucose levels between the control and experimental group were no longer statistically significant. However 60 minutes after the meal, insulin plasma levels were higher in the diaphragmatic breathing group compared to the control group. What can be taken away from this is that practicing diaphragmatic breathing after activities like exercise or a meal for 40 minutes can reduce ROS activity and using diaphragmatic breathing regularly for 15-20 minutes can significantly increase antioxidant levels and increase blood insulin levels in the body.

2.1.1.2 Anxiety and Blood Pressure

Anxiety is “an abnormal and overwhelming sense of apprehension and fear often marked by physical signs (such as tension, sweating, and increased pulse rate), by doubt concerning the reality and nature of the threat, and by self-doubt about one's capacity to cope with it” (Merriam-webster.com, 2018). An example of this would be anxiety caused by the unknown result of a test given by a student at University. Blood pressure is defined as “pressure that is exerted by the blood upon the walls of the blood vessels and especially arteries and that varies with the muscular efficiency of the heart, the blood volume and viscosity, the age and health of the individual, and the state of the vascular wall” (Merriam-webster.com, 2018). Systolic blood pressure is the pressure exerted by blood on artery walls as the heart beats, while Diastolic blood pressure is the pressure exerted by blood on artery walls following a heart beat (Heart.org, 2018). The American Heart Association states that the values for healthy blood pressure are 120 mm Hg and 80 mm Hg for systolic and diastolic blood pressure respectively (Heart.org, 2018).

Blood pressure is a common physiological measurement for evaluating anxiety and stress. Both Chang et al., (2009) and Wang et al., (2010) use blood pressure as a metric for anxiety and stress; the former also measured Visual Analog Scale (VAS) anxiety, skin temperature and oxygen saturation while the latter measured heart activity and blood pressure. The VAS is a questionnaire used to determine the degree of anxiety an individual suffers. Chang et al., (2009) establish in their conclusion that their findings suggest abdominal breathing resulted in a significant decrease of VAS scores and systolic blood pressure. Similarly, Wang et al., (2010) concluded that systolic blood pressure was influenced more than diastolic blood pressure and that the effects can last up to three months. Literature agrees that diaphragmatic breathing brings blood pressures to acceptable, healthy levels as noted above, however the degree to which it does so is inconsistent between the studies.

The explanations for the effects of diaphragmatic breathing on blood pressure and anxiety vary between studies. One of the explanations was that diaphragmatic breathing stimulated the parasympathetic nervous system Chen and Lee (2002 as cited in Chen et al., 2016). The other came from Wang et al., (2010) who state that diaphragmatic breathing resulted in an enhanced baroreflex and improve the vagal nerve tension resulting in an inhibited stress reaction. Breathing has been regarded as a complex process affecting many other processes in the body, so it is likely that both of the explanations given are valid but do not provide the full picture.

The amount of time and duration of breathing training varies significantly between studies. Chang et al., (2009) trained their participants for three days with 5 minute sessions every day. Similarly, Yu and Song (2010) taught an improved version of Mason’s breathing technique to their participants thrice a day for three days. In contrast, Chen et al., (2016) incorporated an 8 week training program with daily practice sessions as well as 30 minute biofeedback sessions to confirm proper technique of diaphragmatic breathing and Wang et al., (2010) organized ten 25-minute sessions twice a week for 5 weeks. for the experimental group whereas the control group performed 20 minutes of breathing daily. From the various articles, five minutes of diaphragmatic breathing every day is sufficient to observe a positive reduction in systolic blood pressure as well as perceived anxiety and results can be experienced after three days of regular practice.

2.1.1.3 Asthma and Asthmatic Symptoms

The duration of interventions in asthma related studies is significantly longer than the other areas discussed. Girodo, Ekstrand and Metivier (1992) performed a 16 week intervention with three trainings a week lasting an hour. Similarly, Grammatopoulou et al., (2011) conducted a 24 week study with 12 sessions spread over the first month of an hour each as well as daily 20 minute sessions at home in the following months. In similar fashion, Thomas et al., (2003) conducted a 24 week study with 75 minutes of training over three weeks followed by daily 10-minute practice sessions for the remainder of the 18 weeks. In contrast to this, Shaw et al., (2009) performed an eight week intervention of aerobic exercise, diaphragmatic breathing and a combination of the two and found that the combination improved Forced Vital Capacity, Forced Expiratory Volume and VO_2 max (maximal volume of oxygen). Forced vital capacity is the maximum amount of air one can expel after a maximal inhalation, Forced Expiratory Volume is the total amount of air one can forcefully expel after a maximal inhalation and VO_2 max is a measure of aerobic capability. Noting that Grammatopoulou et al., (2011) observed the most significant change in their measurements one month after their intervention began, a study regarding the effects of diaphragmatic breathing on asthmatic symptoms should last at least 16 weeks to determine its effect more precisely.

The explanations of the improvement of asthmatic conditions following a diaphragmatic breathing intervention vary in literature. Girodo, Ekstrand and Metivier (1992) reasoned that diaphragmatic breathing strengthened the abdominal, dorsal and oblique muscles and asthmatics benefited significantly from this. Shaw et al., (2009) contradict this by saying that on its own, diaphragmatic breathing does not have the “impetus” or power to improve ventilation distribution, thoracic volume and respiratory muscle strength which aerobic exercise and a combination of diaphragmatic breathing and aerobic exercise did. It should be noted however, that the intervention of Shaw et al., (2009) only lasted eight weeks and thus might not have provided enough evidence to the contrary. Grammatopoulou et al., (2011) observed that diaphragmatic breathing decreased respiratory rates, symptoms of CO_2 levels in the blood known as hypocapnia and increased End-Tidal CO_2 levels thereby reversing airway bronchoconstriction. In summary, diaphragmatic breathing strengthens abdominal muscles that aid in breathing, reduce breathing rate and lung-airway constriction. However, it needs to be practiced over an extended period of time to have a noticeable effect.

A diminution of the benefits of interventions consisting of breathing retraining were found in a two studies. Girodo, Ekstrand and Metivier (1992) had participants undergo a 16 week training program where they trained for an hour 3 times a week. They followed up two months after the study and found that many participants had relapsed into old habits and medication levels. They observed that adherence to treatments requiring long term commitment can be aided by individually tailoring the progression and training of diaphragmatic breathing and encouraging involvement of other physical activities. This was echoed by Thomas et al., (2003) where they found that the changes in the Quality of Life scores of their participants were only visible in half of the sample after 1 month and in a fourth of the sample after 6 months. Both studies were long term (> 4 weeks) and it seems that these interventions are helpful but not practical due to commitment issues.

In conclusion, the amount of time diaphragmatic breathing needs to be practiced varies between the contexts of its use. In the case of oxidative stress, the time described in literature ranged between 15 minutes to an hour. The breathing sessions were done after activities that induced an increase in oxidative stress such as strenuous physical activity or the consumption of a meal. The effects of an increase in melatonin and reduction in cortisol due to diaphragmatic breathing were immediate however temporary (diminishing within 24 hours).

In contrast, the effects of regular diaphragmatic breathing can lower systolic blood pressure and can be experienced for up to three months after training in the context of anxiety and blood pressure. The training however, needs to be done regularly, once or twice a week according to literature.

In the case of asthma and reducing asthmatic symptoms, diaphragmatic breathing needs to be practiced extremely regularly for extended periods (longer than 4 weeks). This level of commitment makes this treatment inconvenient and impractical for many and requires personalized motivation plans and tailored progress schemes to be truly effective.

2.1.2 Measurement

Literature on the use of RIP for measurement in experiments was approached with three aspects in mind; the placement of the bands on the participants, the observed causes for measurement errors and what RIP was compared to and its subsequent evaluation.

There is little variation in the positioning of the bands between studies. Retory et al., (2016) placed the thoracic band under the armpits, a region known as the axilla. Courtney, Van Dixhoorn and Cohen (2008) employ the LifeShirt™ for measurement in their study, which has bands sewn into a shirt placing the thoracic band under the axilla and the abdominal band around the abdomen, a slightly unclear description. Caretti et al., (1994) had vague descriptions of the positioning of the thoracic band being placed around the ribcage and the abdominal band around the abdomen. Mehra and Strohl (2009) had a more detailed description, described the placement of the thoracic band at the level of the nipples and the abdominal band at the umbilicus. Interestingly, there was no medical protocol followed or quoted in any of the studies. In addition, it was not possible to find details of RIP usage in hospitals or other medical settings. From literature it seemed that every study had its own position for the bands which generally matched, however some studies mention the slight redundancy of exact placement due to pre-calibration of the system before measurement.

Literature agrees on the limitations of RIP in the context of moderate to high amounts of physical activity. Retory et al., (2016) used RIP to evaluate ventilatory adaptation during mild physical activities. They concluded that a single band surrounding the thoracic region in combination with a nasal pressure signal was a useful tool for monitoring ventilation in mild, submaximal physical activity. This agrees with the conclusions of a study using RIP to measure exercise tidal volumes (Caretti et al., 1994), where the researchers noted that RIP could replace turbine flow meters at work rates below 180 Watts in activities requiring low amounts of body movement. For measurement purposes the activities performed should not involve high amounts of physical activity. Mayer et al., (2000) conducted their measurements as the participants lay in the supine position (for detection of apnea). Similarly, Courtney, Van Dixhoorn and Cohen (2008) instructed their participants to sit in 3 different positions: normal, erect and slumped while breathing in 3 different ways: normal, thoracic and abdominal. These conditions required no exertion or cause for muscular interference. In order to reduce the amount of errors and interference, activities involving participants to be seated, lying or standing are preferable to those involving moving such as running or cycling.

The causes of error during RIP measurement differ slightly in literature. Both Caretti et al., (1994) and Mehra and Strohl (2000) mention large amounts of movement and slippage to be the two factors that contributed most to errors in measurement. The former attempted to stop the slippage by using canvas straps around the band however this did not prove effective. Courtney, Van Dixhoorn and Cohen (2008) found that the RIP was ineffective in measuring vertical expansion of the thoracic region in the case of spinal extension and did not account for thoracic abnormalities or scoliosis whereas Manual Assessment of Respiratory Motion (MARM), the method they were comparing RIP against, did. Furthermore, Caretti et al., (1994) found that exercises that increased pulmonary blood volume and oscillated with respiration caused significant errors in RIP tidal volume measurement. Research on the topic of RIP measurement introduces many sources of errors to be aware of although the setup of an experiment makes it more vulnerable to certain sources of error.

The RIP has been validated against various other forms of tidal volume measurement. It was compared to the nasal pressure recorded by a pneumotachometer where the conclusion was that with appropriate time domain filtering, and calibration, thoracic RIP would be sufficient in routine practice that accounts for mild displacement of the bands (Retory et al., 2016). In the context of MARM (Courtney, Van Dixhoorn and Cohen 2008) , it was proven superior to RIP however was deemed impractical due to the subjectivity of the measurement, training required for proper measurement and lack of independence this method of measurement afforded. RIP was compared to turbine flow meters (Caretti et al., 1994) where the conclusion was that it was accurate if no strenuous physical activity was performed. In contrast Mayer et al., (2000) used polysomnography to acquire nasal End-Tidal CO_2 volume and nasal oral thermistor signals and concluded that generally, their RIP based solution was able to detect episodes of apnea however mention that a large number of episodes were not detected due to movement and obstructed breathing. Mehra and Strohl (2009) provided an overview of the techniques used to monitor and evaluate respiratory function and concluded that RIP was “an acceptable method for the semi-quantitative measurement of ventilation”. In summary, the RIP has been compared to a number of measurement techniques and for situations involving stationary positions with little movement and strenuous activity, the non-invasive nature and accuracy of RIP makes it a viable choice.

There are a number of calibration methods for RIP, some of which are applicable to a larger amount of situations than others. Chadha et al., (1982) reviewed the Isovolumetric calibration method, the Least Square method and the method of Stagg and associates. In their conclusion, they state the recognition that the relative change in pressure and volume characteristics of the rib cage and abdomen at different lung volumes might change the relative calibration and slope of the isovolume angle. This also meant that this calibration method was prone to errors when there is a change in body position. Therefore, the system would need recalibration for every change in body position, thus rendering it impractical. They discuss that with the Least Squares Method (LSQ), it is possible to measure tidal volumes with a good degree of accuracy in the context of another experiment (Abraham et al., 1981). The deviations from spirometry for validation trials in standing and supine positions were insignificant enough to prove that the magnitude of regions moving independently of the grouped rib cage and abdominal compartments is negligible. They state that LSQ calibration of RIP is satisfactory for human patients.

In conclusion, RIP has been validated as a method to accurately measure respiratory flow under the right circumstances. Literature also covered the importance of calibration as it lessens the importance of exact positioning of the bands on the subject. Situations where the participants are performing activities that require them to move a lot should be avoided as one of the biggest causes for error in measurement using RIP is slippage of the bands that nullifies calibration done beforehand. Literature tends to agree to a certain degree on the placement of the bands: some sources place the abdominal band at the level of the navel or umbilicus while others place it lower around the waist. As for the thoracic band, some sources place it directly across the chest under the armpits or axilla whereas others place it at the level of the nipples.

Sources also point out the shortcomings of RIP as a measurement method due to its susceptibility to error from physical activity, a situation of interest in many studies. Researchers noted that the RIP does not measure vertical expansion of the thoracic region during spinal extension, only the lateral expansion. In addition, it does not account for individual suffering abnormal thoracic growth or scoliosis. These factors should be given attention during the testing phase of the project where participants are screened for the testing. Another important observation is that asynchronous breathing introduces errors in the classification and was a significant issue in studies that used untrained participants.

That being said, the level of freedom RIP affords participants and its low level of intrusiveness compared to other methods such as pneumotachography, polysomnography and spirometry that it has been validated against, makes it a feasible choice for a study of this scope.

2.1.3 Classification

This section will introduce the two kinds of classification and give explanations and examples of a few classifiers. Literature on classification is then reviewed to investigate the types of classifiers used in the context of respiratory data, the parameters used and the presence of signal preprocessing.

2.1.3.1 Types of Classification

There are numerous classifiers, each unique in its functioning and exemplary in certain situations. Classifiers can be categorized as Linear or Non-linear, depending on whether the classes can be separated by a linear plane or not. Figure 2 shows an example of a linear separation of classes, where the classes are easily distinguished with a straight line. Figure 3 shows a problem where the classes cannot be separated with a line and requires a different approach.

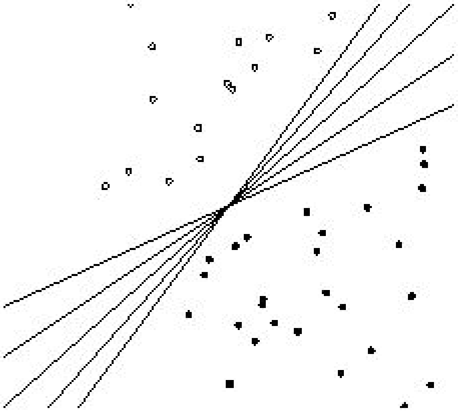


Figure 2.1: Linear separation of classes (nlp.stanford.edu, n.d.)

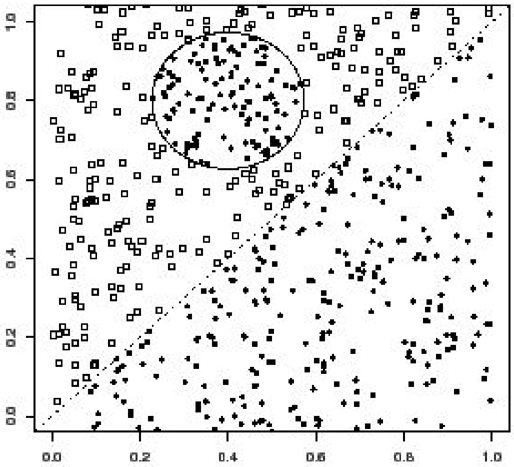


Figure 2.2: Non-linear separation of classes (nlp.stanford.edu, n.d.)

For some more background on linear and nonlinear classifiers, two of each will be introduced in the sections below.

2.1.3.1.1 Linear classifiers

Linear Discriminant Analysis

Invented by a man named R.A.Fischer in 1936, Linear discriminant analysis looks for the optimal way to linearly combine variables that best predicts the separation of two classes. Fischer developed a score function that is then used to estimate the linear coefficients that maximize the outcome of the score function i.e. the score. A parameter known as the Mahalanobis distance is calculated that characterizes the distance of separation between two classes; the higher this value, the more the two classes are distinguishable from each other (Chem-eng.utoronto.ca, 2018). This type of classification uses statistical values such as averages, covariance matrices and class probabilities. An example of the use of a linear classifier can be seen in Figure 2.

Logistic Regression Analysis

Logistic Regression Analysis is another statistical method used for analyzing data where there is more than one independent variable that determines the outcome. This outcome is dichotomous in nature, meaning it only has two possible outcomes; yes or no, alive or dead, diaphragmatic breathing or not diaphragmatic breathing. Logistic Regression Analysis is used to find the best fitting model to describe the relationship between a dichotomous variable of interest and a set of independent variables. The coefficients, and their weights and significance levels are all generated to predict the probability of the presence of the variable of interest (Schoonjans, 2018). This probability is the outcome of a Logit Transformation given by the following equations:

$$odds = \frac{p}{1-p} = \frac{probability\,of\,presence\,of\,characteristic}{probability\,of\,absence\,of\,characteristic}$$

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

2.1.3.1.2 Non-linear classifiers

Artificial Neural Networks

Artificial Neural Networks (ANNs) as described by their inventor are: "... a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs." (Pages.cs.wisc.edu, 2018). ANNs are a programming paradigm that are modeled after the neuronal structure of the human brain, albeit on a smaller scale. A neural network, as the name implies, is a network made up of nodes called neurons like in the brain. These neurons are connected to each other via "connections" which each carry their own weight based on the importance of the node or input they are linked between. Between the input layer and the output layer, there exist a single or multiple hidden layers where the processing is done using the weighted connections (Pages.cs.wisc.edu, 2018), as shown below in Figure 2.3. The inputs have a weight applied to them and these values are summed in the connected node. If this sum exceeds a certain value, the neuron "fires" and the value is sent to the next layer with a weight applied to it.

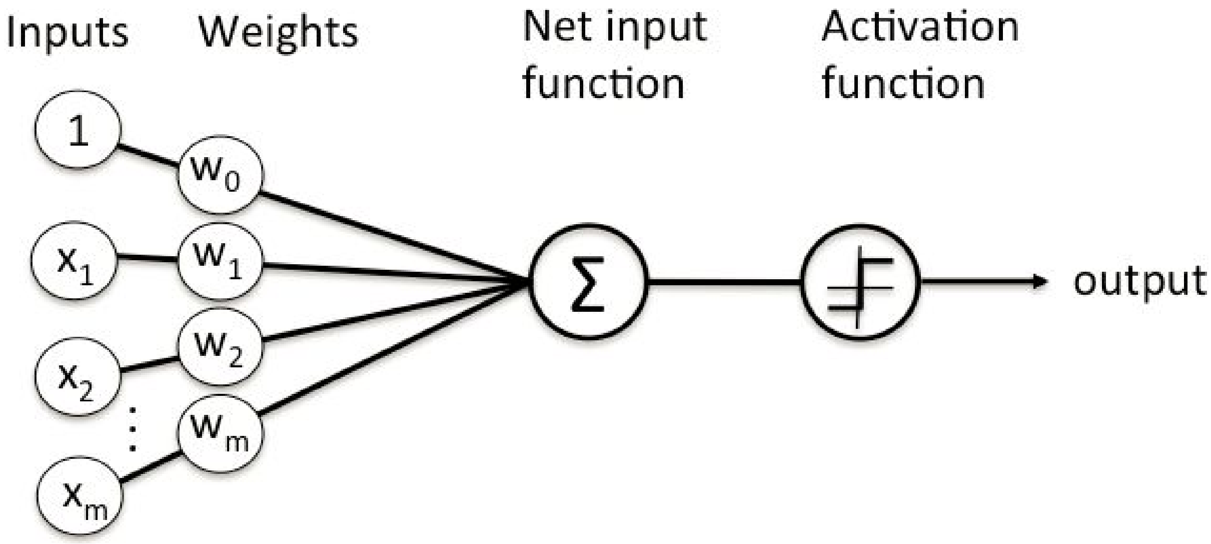


Figure 2.3: *Representation of an ANN (Chris V. Nicholson, 2018)*

An ANN learns through supervised learning where in the beginning, it makes a guess as to what the pattern might be, is then corrected and adjusts its weights accordingly (Pages.cs.wisc.edu, 2018).

Support Vector Machines

A Support Vector Machine is a supervised learning method that maps input-output functions from labelled training data. For the purpose of classification, nonlinear kernel functions are used to transform input data into a high-dimensional feature space in which the data becomes more separable compared to the original input space. Maximum-margin hyperplanes are then calculated (Wang, 2005). A nonlinear kernel function is a mathematical method to apply a linear relationship to nonlinear settings by mapping data to a higher dimension where it displays linear properties (Rai, 2011). An example would be considering the following problem shown in Figure 5.

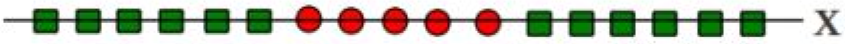


Figure 2.4: *One dimensional classification problem (Rai, 2011)*

This problem can be solved by using a nonlinear kernel function that transforms the input data into a higher-dimensional feature space as shown in Figure 6.

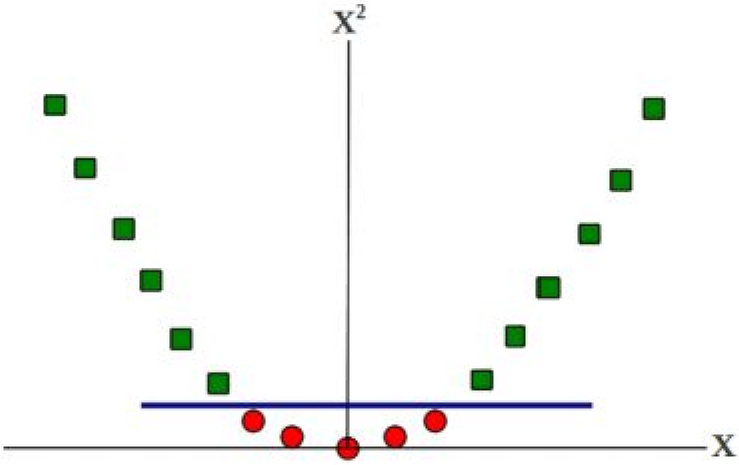


Figure 2.5: *One dimensional problem transformed into a higher dimensional problem (Rai, 2011)*

From the representation in Figure 6, it is now possible to find a linear relationship in the data otherwise impossible from the previous representation in Figure 5.

2.1.3.2 Literature on Classification

When looking at literature regarding the classification of breathing signals, three factors are of importance: the type of algorithm being used for classification, whether or not any signal preprocessing is done and if so what kind, and lastly the features that were extracted to train the classifiers.

The type of artificial intelligence used for classification of breathing signals differs in literature. Lee et al., (2012) used a Neural Network to differentiate between normal and abnormal breathing patterns. Raoufy et al., (2016) and Raoufy et al., (2012) employ Neural Networks to differentiate between controlled and uncontrolled and non-atopic and atopic asthma, and to model respiratory volume based on thoracoabdominal breathing movements respectively. The latter study justified the use of a Neural Network as the study used RIP to measure respiratory movement and found that it handled asynchronous breathing (a cause for error in measurements) better than a linear classifier could. Garde et al., (2012) identified periodic and non-periodic breathing patterns in individuals suffering Chronic Heart Failure (CHF), using both linear and non-linear classifiers (Support Vector Machines) and compared their classification performance. It appears the type of classification used depended on the way the data was measured and the types of features that were extracted from the signal.

A couple of studies utilize the flow envelope signal for feature extraction and classification. The envelope of a signal is the boundary within which the signal is contained, when viewed in the time domain. Its an imaginary line that can be modelled based on the characteristics of the signal (Envelopes, n.d.). Garde et al., (2008) conclude in their study that as a preliminary study, the results evoke consideration of flow envelope signal analysis to characterize different respiratory patterns in CHF patients. In a similar study by Garde et al., (2010), they state in their conclusion that envelope flow signal analysis “can be a useful tool to characterize different respiratory patterns in CHF patients and healthy subjects.” Both studies agree on the promising utility of flow envelope signal analysis in the context of CHF patients however its reliability may vary in other contexts.

The features extracted from data varies significantly between studies. Lee et al., (2012) extracted a number of features from the respiratory motion signal; standard deviation, autocorrelation max value and delay time, acceleration and velocity variance, breath frequency, max power of fourier transform, multiple linear regression coefficient, maximum likelihood estimates and principal component analysis. Raoufy et al., (2016) extracted peak to peak interval and the amplitude of the peaks, similar to Garde et al., (2010) who extracted frequency interval and modulation frequency peak. Various parameters can be used for classification and the accuracy of the classification may depend on the number of parameters.

Signal preprocessing to filter out noise and other artefacts isn’t always performed in literature. Lee et al., (2012) mention no processing done to filter out artefacts or noise. Similarly, Raoufy et al., (2012) employed no artefact reduction or frequency filtering. On the other hand, Raoufy et al., (2016) manually observed the participants during data collection to mark in time when a participant moved, coughed, sneezed or otherwise disturbed the measurement. Garde et al., (2010) filtered out data points that fell outside the 1st and 99th percentile of the signal’s amplitude and the gaps were interpolated based on Autoregressive (AR) signal modeling. In addition, low pass filters were applied to the signal as the breathing frequency did not exceed 0.5 Hz. Similarly, Garde et al., (2008) used a polynomial filter to get rid of medium-high frequency noise leaving only the low frequencies. From literature, when signal preprocessing is performed, outliers are removed and medium to high frequencies are filtered out.

In conclusion, the types of classifiers used for breathing pattern recognition vary between studies. The merit of a non-linear classifier such as an Artificial Neural Network being able to deal with data from asynchronous breathing was mentioned in literature, and was used in a number of studies found. Other studies compared the accuracy of linear and non-linear classifiers based on a set of parameters. The type of classifier used depended on the complexity of the data being recorded and the goal of the classification. The features extracted from the data were loosely dependent on the goal of the studies. Some studied focused more on time domain components while others on those from the frequency domain. The analysis of the envelope signal of a respiratory flow signal was found to be a promising tool for classification in the context of Chronic Heart Failure patients. This project will focus on simple linear classifiers such as linear discriminant analysis and logistic regression analysis as the outcomes of classification are dichotomous (diaphragmatic breathing or not).

Signal preprocessing was done primarily in the studies using the envelope signal of a respiratory flow signal. Low pass filters and polynomial filters were used to remove medium-high frequency artefacts and outliers falling outside the 1st and 99th percentile were removed. A more straightforward and practical, though time consuming approach was to observe the participants during measurement to mark in time when artefacts were introduced due to movement etc. however this method is only feasible in studies with small sample sizes.

2.1.4 Habit forming

Literature on habit forming was explored to better understand how a habit is formed and how long it takes for a habit to form. This will allow for more informed decisions during the ideation and methodology phase of the project when it comes to forming the habit of breathing diaphragmatically every day.

Literature on habit formation is extensive and agrees on the how a “habit” is formed. Gardner, Lally and Wardle (2012) state that habits are actions triggered automatically in response to contextual cues that have been associated with their performance, and that the mere repetition of a simple action in a consistent context leads to the action being activated upon being exposed to said contextual cues. Gardner et al., (2014) echo this explanation, stating that a habit is a learned process that generates an automatic response to contextual cues. Gardner (2012) uses the term automaticity, which is the ability to do something automatically without much conscious thought, and expresses the necessity to differentiate between cue-dependent automaticity from context-dependent automaticity. He reflects that a habit is better conceptualized as a form of automaticity which once formed, need not be defined by frequency performance. Therefore, a habit is formed by the repeated performance of an action to a certain stimulus or context.

The amount of time it takes for a habit to form depends on several factors and varies greatly in literature. Gardner et al., (2014) performed a study evaluating a habit-based dietary behaviour change intervention and found that it took only two weeks to form a habit, evaluated using questionnaire, which was either maintained or had increased further upon follow up. Gardner and Lally (2012) found that the duration of time it took for a habit to form was dependent on the individual’s personal motivation to form said habit. In addition, they also found that relative autonomy had a direct effect on habit strength above the influence of past behavioural frequency. Their results showed that it took an average of 91 days for automaticity to plateau, however the interquartile range was 74 days indicating a large difference for individuals. Lally et al., (2009) conducted a study to incorporate a new habit in its participants and found that it took an average of 66 days for the habit to develop however, the range was from 18 to 254 days also indicating a large difference in time it took for habits to develop on an individual basis. The habits in the experiment ranged from drinking water more regularly to exercise every day. In summary, the amount of time it takes for a habit to form is based on an individual's motivation to develop the habit, the autonomy they are given to practice the activity and the complexity of the activity.

A habit can be lost if not performed frequently enough. Tobias, (2009) found that in his experiment to change behaviour using memory aids, there was a decrease in habit strength caused by the reaction to the memory aid being performed less frequently. Maintaining a habit is done by performing it frequently in order to strengthen it as a response. Judah, Gardner and Aunger (2012) attempted to form the habit of flossing and found that their habit scores were significantly lower eight months after the intervention compared to four weeks after it. Literature has found that a habit can weaken and decay over time if not performed frequently.

Habits are formed when an action is repeatedly done as a response to a certain cue or context. The amount of time a habit takes to form varies significantly between individuals and is dependent on their motivation to form the habit and the independence they are given to form said habit. Habits can weaken over time if not performed frequently and is a point to reflect upon when weaning users off the device to perform diaphragmatic breathing on their own.

2.1.5 The importance of feedback

Learning a new skill can take time if done without the aid of feedback from an experienced individual. The type of feedback given to people learning diaphragmatic breathing is of interest as it is informative for procedures during the data collection later on in the project. Literature was reviewed looking for the types of feedback used for teaching diaphragmatic breathing.

Feedback is an important part of the learning process for diaphragmatic breathing and has been provided in a couple of ways in literature. Ma et al., (2017) employed an experienced yoga instructor to teach participants diaphragmatic breathing. During the session, the breathing waveforms were displayed on a screen visible to the yoga instructor who would then adapt his instructions based on the waveform of the participant. In a similar fashion, a study into diaphragmatic breathing to treat chronic migraines by Kaushik et al., (2005) used a trained yoga instructor to teach their participants diaphragmatic breathing. The respiration was displayed on a monitor visible to the instructor and they made changes to their instruction as such. These two studies used a form of live feedback but not directly to the participant but via the instructor. In contrast, Wang et al., (2010) used an EMG signal to give live feedback to the user where the color of the signal would turn green if the abdomen was being employed during the breathing. They used gamification to motivate the participants to change the color of the wave from yellow to green by utilizing more of their abdomen. Unfortunately only the study by Wang et al., (2010) compared the use of biofeedback with the experimental group as opposed to no feedback with the control group and found that biofeedback assisted diaphragmatic breathing had a greater impact on blood pressure.

2.2 Expert Interviews

After looking through literature, three separate interviews were conducted with an expert practitioner of Tai-Chi and acupuncture, Parviz Sassanian; a voice and singing coach, Ineke Ter Hedde and a researcher at the Biomedical Signals and Systems faculty at the University of Twente, Ainara Martinez Garde. The questions prepared can be found in the Appendix 11.A. In every interview, the questions that were prepared were abandoned through the course of the interviews which took a more free-flowing form that touched upon the themes of the questions.

2.2.1 Expert Interview with Parviz Sassanian

Parviz Sassanian has studied Chinese medicine and Tai-Chi in Utrecht and runs a practice in Enschede. Mr. Sassanian described breathing as a tool that is used to aid actions and can be used to change one's emotional state and even personality, given enough time. Breathing applies to the mind, body and spirit and bad, or dysfunctional breathing, leads to a lower quality of life.

He has used diaphragmatic breathing to treat conditions such as eczema and diabetes but spoke about how it can be used to fight ageing and heal others. Mr. Sassanian mentioned that science has not developed enough to explain why diaphragmatic breathing and acupuncture work the way they do and it is difficult to pinpoint what happens because science only observes results and obtains statistical data in its search for mechanical and or linear relationships. He spoke about how the body functioned as a quantum system, and thus why modern science fails to grasp the understandings of the human body on a more holistic level.

He observed that women, more than men, were used to employing more of their thoracic region for breathing rather than their abdominal region and employ "Reverse Breathing" (abdomen contracts on inhalation and expands during exhalation). Reverse breathing was introduced as a negative phenomenon as the abdomen is contracted during inhalation allowing less oxygen into the lungs.

For the purpose of training, he related proper diaphragmatic breathing to the expansion of a ball in the abdomen while inhaling; one that expands the front, back and sides of the abdominal region. He warned however that breathing through the diaphragm generates a lot of "chi", the force of life in the Chinese medical system, and that if not prepared correctly through training, the body is incapable of dealing with this energy and it can cause damage. Therefore, he advised to practice it for around 10 to 15 minutes every day.

2.2.2 Expert Interview with Ineke Ter Hedde

Ineke Ter Hedde has been teaching classical singing for the past 30 years. Ineke has developed a method where she listens to one's breathing and can pinpoint the issues with their technique. While coaching amateur singers and found that many of them, due to bad habits, maintained a lot of tension in their chest and abdomen when breathing in deeply. She says the mere awareness of the tension can help correct their technique and that total relaxation can even trigger extreme emotions like laughter or tears. She estimates we only use around 30-70% of our lung capacity when breathing in "deeply" due to bad habits formed earlier.

Ineke explained that proper diaphragmatic breathing should make the diaphragm sink to allow more air into the lungs, and contract to push out air. Breathing should cause the abdominal region to expand completely but strain on the back and subconscious muscle stiffness can restrict this. Some of her students have experienced a reduction in asthma and asthmatic symptoms, back pain and emotional related stress however after the span of around a year. She suggested that the RIP bands should be placed around the hips for the abdominal region and under the pectoral muscle for the thoracic region.

The interviews with Mrs. Ter Hedde and Mr. Sassanian show an agreement on the general benefits of diaphragmatic breathing and that some benefits take a significant amount of time to manifest. These views are also in agreement with the conclusions from the literature research on physiological health benefits in Section 2.1.1.

2.2.3 Expert Interview with Ainara Garde

Ainara Martinez Garde currently works as a researcher at the University of Twente and in the past has worked on applications and solutions involving biosignals. Previously, Ainara had worked on a game with children where they had an accelerometer strapped to their ankles to measure their movement and none of them, with the exception of one child, found it to be annoying or irritating. With regards to projects she worked on involving breathing measurement, she said to look at inspiratory, expiratory time and possibly peak to peak time but not amplitude as it could vary depending on the placement of the bands and slipping.

Ainara recommended that for the scope of this project, the signals from the two bands could simply be summed and then the contribution of each band could be normalized and expressed as a percentage. After setting a threshold values for the percentages at which point thoracic breathing changes to abdominal breathing or vice versa, classification can begin. She recommended either summing chunks of the signal of periods of around 1 minute or checking individual inspiration and expiration pairs of the signal.

For signal preprocessing, Ainara recommended using a band pass filter to remove low frequency baseline drift and medium to high frequency noise from talking and movement. She mentioned that the type of classification depended on what was being recorded. She also mentioned during the data collection of her previous experiments, some chunks of the signal had so many artefacts that they were omitted entirely.

For classification, Ainara suggested using a linear discriminant or logistical regression analysis that gives a probability of using a type of breathing or not with a threshold to distinguish between the two. She also mentioned doing frequency analysis of the envelope signal as well as normalizing the two signals and finding the contribution of each signal to the sum to then use for classification.

2.3 Market Research

After doing research into literature and interviewing experts in the field of breathing, training and measurement and classification, an analysis of the current technologies available on the market was necessary to compare the novelty of the solution. Research was done into the different types of products and/or services available on the market for breathing training. The search results listed breathing training apps and breathing training devices, some of which taught diaphragmatic breathing. It was difficult to find information on the classification method used in the apps if any, due to its proprietary nature. The technology was sorted into the categories of wearables and devices, and smartphone applications.

2.3.1 Wearables and Devices

2.3.1.1 Spire: Stone and Health Tag

Spire is a company that produces wearable devices that track biometric data to evaluate breathing, sleep and activity. The Spire Stone, shown in Figure 2.6, sits on the waist and uses force sensors to monitor activity and measures the expansion and contraction of the abdomen during inhalation and exhalation. Their algorithm uses real time classification that takes around 4-5 minutes to completely work and uses features such as: inhalation exhalation duration and slop, hold durations, inhalation/exhalation ratio, consistency and the morphology of the signal (Spire, 2018). The Spire Health Tag, shown in Figure 2.7, is stuck on the inside of a bra or the waistband of boxers and uses similar technology to the Spire Stone.

From their website it is unclear if their app teaches diaphragmatic breathing, however they do have an article with instructions on how to perform diaphragmatic breathing on their blog. On their blog, they even mention that the parasympathetic fibers are in the vagus nerve that is relaxed during diaphragmatic breathing agreeing with literature on the mechanism of relaxation (Spire, 2018).

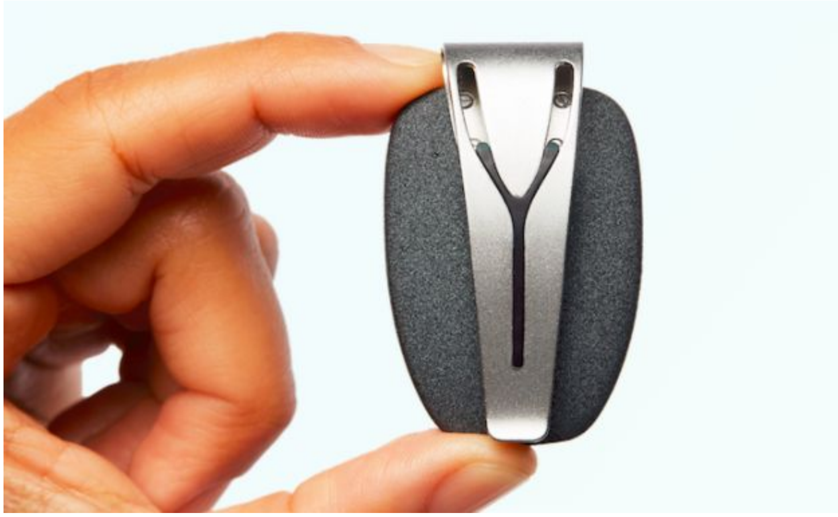


Figure 2.6. *The Spire Stone, a clip-on wearable* (Spire, 2018)

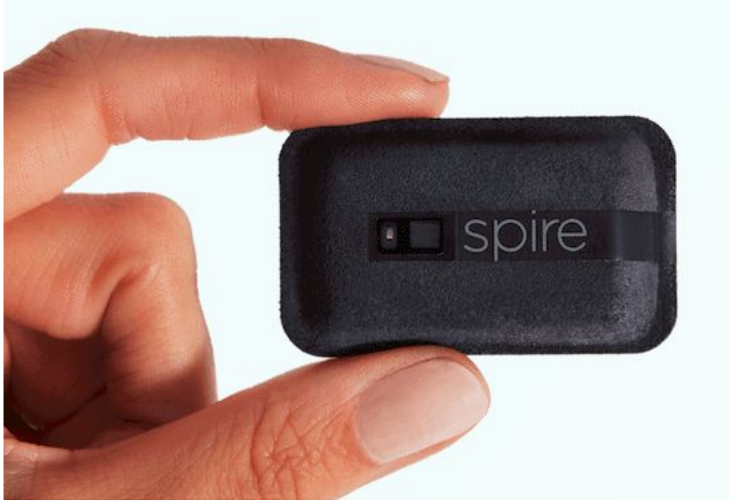


Figure 2.7. *The Spire Health Tag* (Spire, 2018)

2.3.1.2 Prana

Prana is a company whose wearable device (Figure 2.8) sits on the belt, much like the Spire Stone, and measures breathing and posture using an accelerometer and a displacement sensor. The platform evaluates breathing patterns and takes into account the effects of posture on breathing as well as differentiates between chest breathing and diaphragmatic breathing (Prana.co 2018). Prana senses the angle of the diaphragm and the posture with the accelerometer as well as the breathing with the displacement sensors and uses this data as input for training games (Figure 2.9) that it offers in its accompanying smartphone application. It sends the user notifications on their smartphone in real time to tell them to sit straight and breath more with their diaphragm.

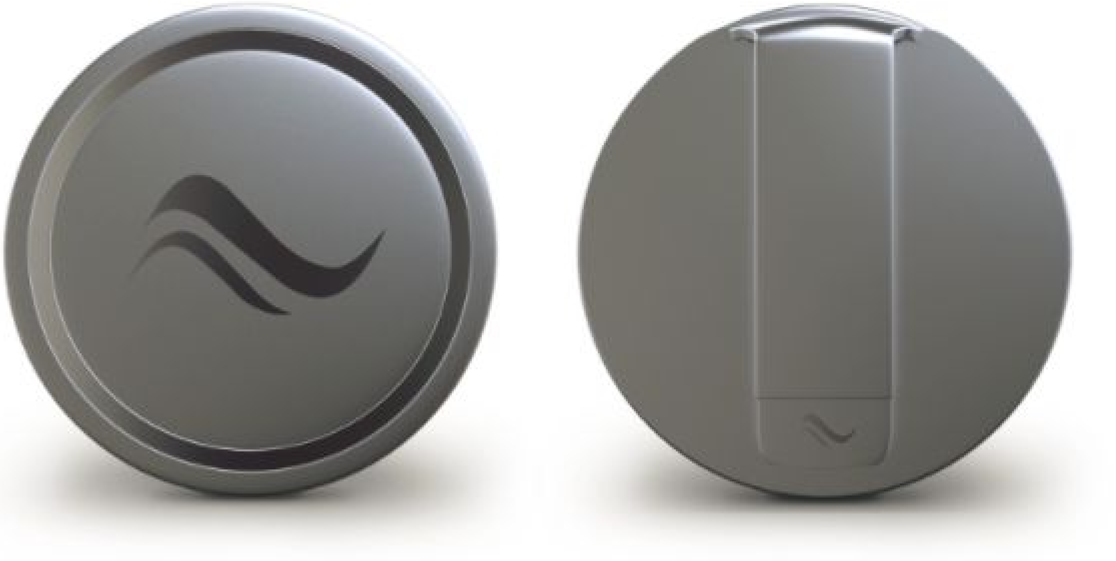


Figure 2.8. *Prana Wearable Device shown on the homepage of the website (Prana.co, 2018)*

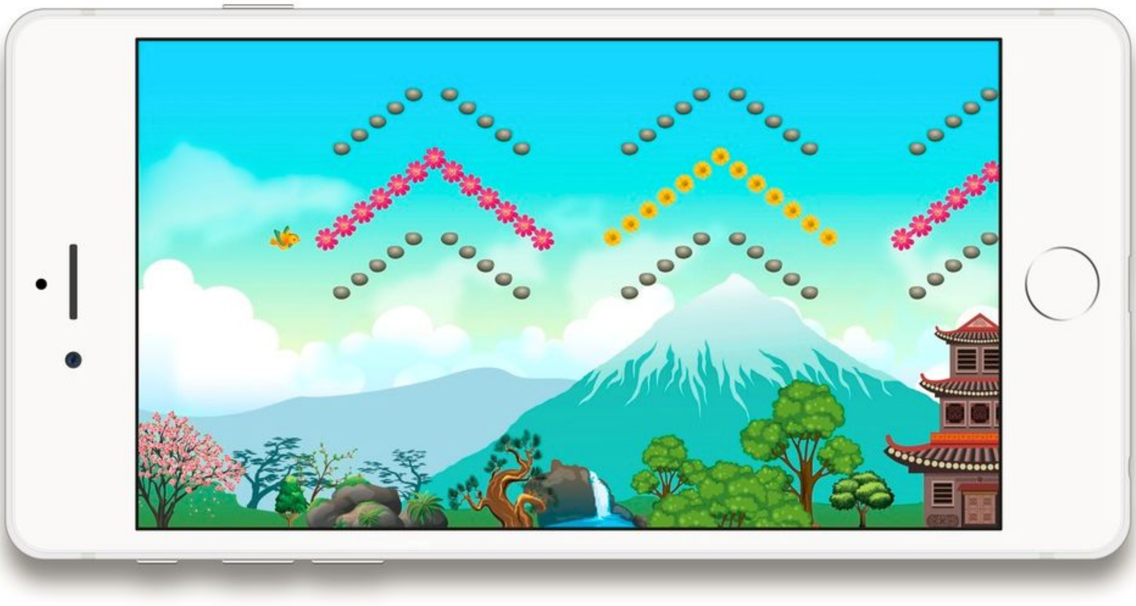


Figure 2.9. *Android game to help teach and practice diaphragmatic breathing used in combination with the Prana (Prana.co 2018)*

2.3.1.3 Focusband

A band (Figure 2.10) embedded with sensors that is worn around the head and connected to a smartphone via bluetooth, it measures frequencies (Figure 2.11) of the electrical brain activity on the forehead and can classify the frequencies in Image 7 below. The Focusband uses Electroencephalography (EEG) with its three silver oxide sensors to do this (FocusBand 2018). The accompanying app provides instructions on how to breathe with animations and measures the frequency of electrical brain activity. Their website lists a suite of apps available for neurofeedback for people suffering from ADD, anxiety or depression. (FocusBand 2018), mental training tools for sports and apps that help maintain a state of reduced stress and a sense of calm.

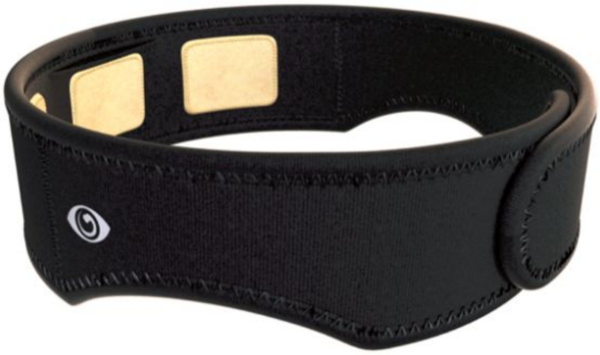


Figure 2.10. *The Focusband wearable shown on the website homepage (FocusBand, 2018)*

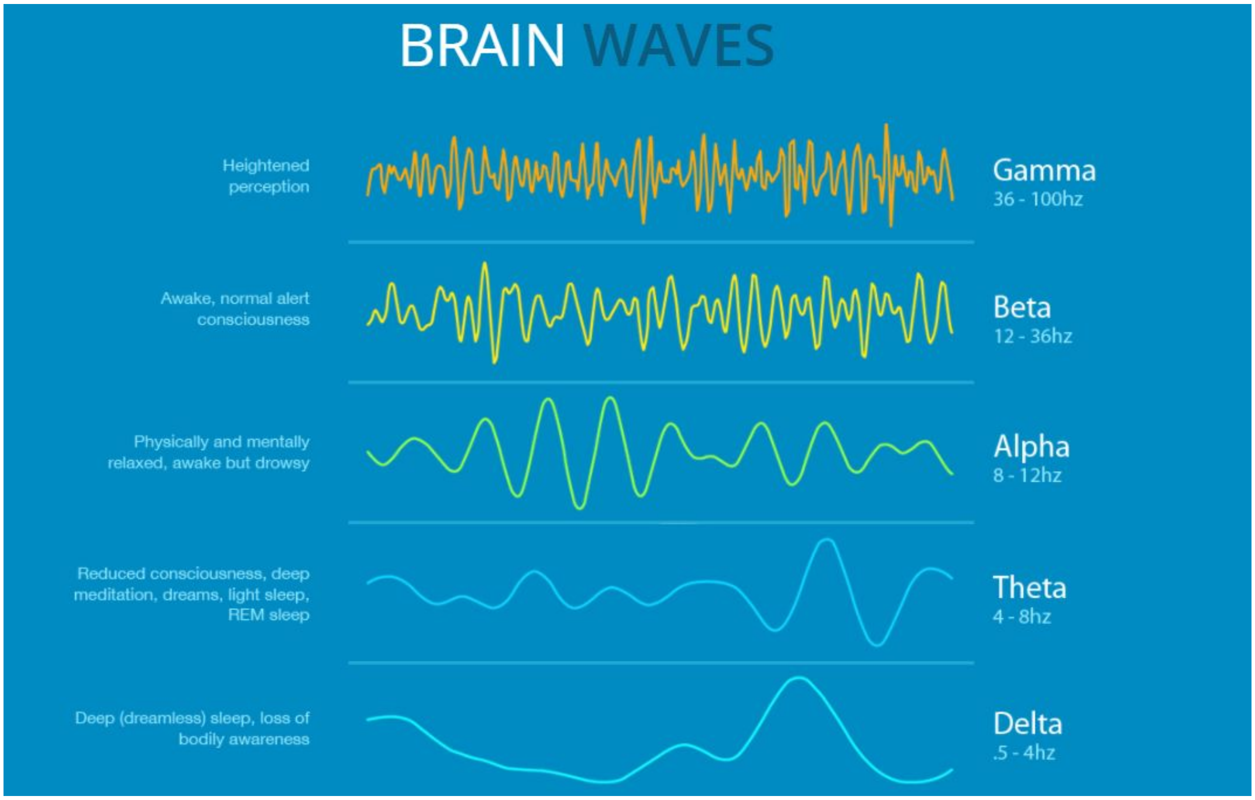

BRAIN WAVES
Heightened perception
Gamma
36 - 100hz
Awake, normal alert consciousness
Beta
12 - 36hz
Physically and mentally relaxed, awake but drowsy
Alpha
8 - 12hz
Reduced consciousness, deep meditation, dreams, light sleep, REM sleep
Theta
4 - 8hz
Deep (dreamless) sleep, loss of bodily awareness
Delta
.5 - 4hz

Figure 2.11. *The various frequencies the FocusBand measures (FocusBand, 2018)*

2.3.2 Smartphone Applications

2.3.2.1 Breathe2Relax

This app was developed by the National Center for Telehealth and Technology in the United States. It Marketed as a stress management tool providing information on the effects of stress on the body, it contains exercises as well as instructions on how to perform diaphragmatic breathing. The app (Figure 2.12) features timed inhalation and exhalation sessions that can be customized according to user capability. The user can record their perceived stress levels before and after performing the exercise to log the change in mood (Youtube. 2018).

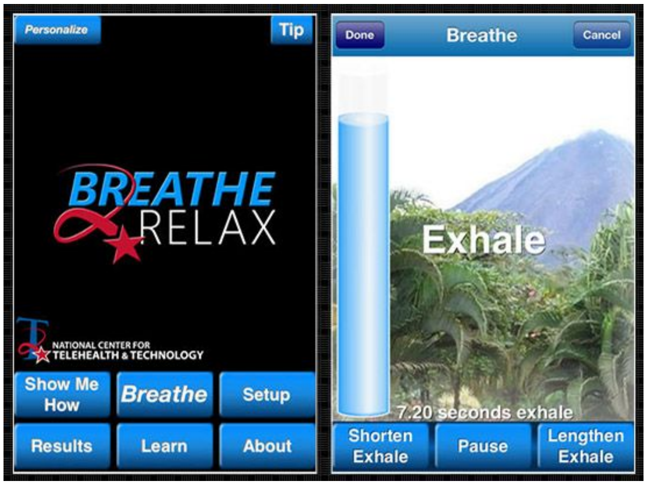


Figure 2.12: *Screen grabs from the application of the homescreen and the breathing exercises (Topapps.net, 2018)*

2.3.2.2 BellyBio

An app available exclusively for the iPhone, BellyBio (Figure 2.13) uses the accelerometer to measure and provide feedback for the training of diaphragmatic breathing. Dr. Kendall Ho (Youtube. 2018) explains how user is instructed to lie flat or lean back against a surface and place the phone on their abdomen. The app plays sounds of waves as cues for the user to breathe in and out. If the phone tilts away from the user while they breathe in, and towards the user while they breathe out, the app registers that they are successfully performing abdominal breathing and starts playing music. It also displays a blue and red ball for inhalation and exhalation respectively.

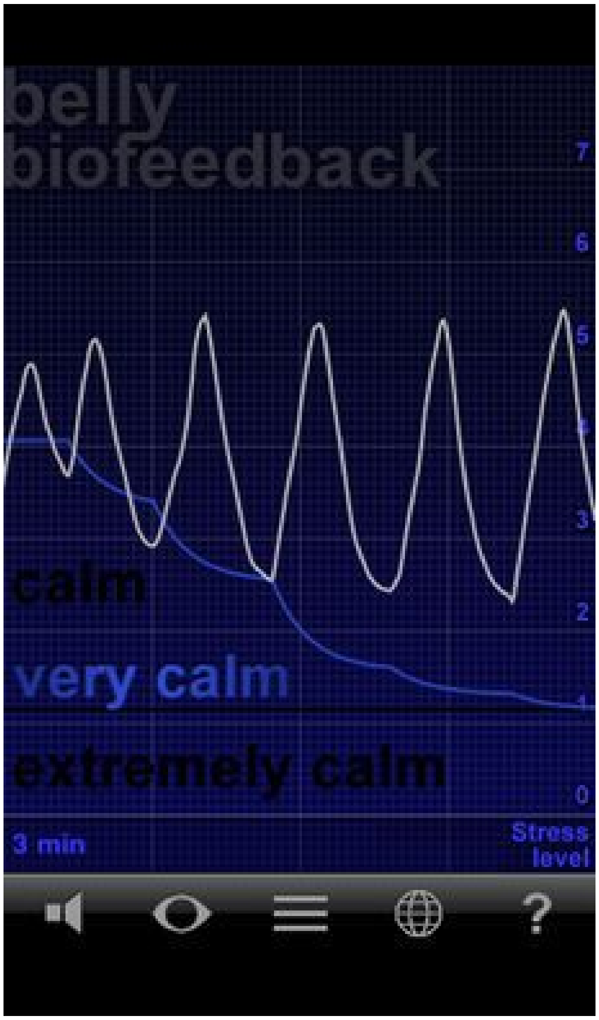


Figure 2.13: *Screenshot of the main screen of BellyBio (App Store, 2018)*

2.3.2.3 Box Breathing

This app (Figure X) is a breathing training app that is meant to train people to use diaphragmatic breathing however there is no form of feedback and the interface only uses a timer, audio prompts and animations as cues for the user. The founder, David de Souza, said that a useful technique he discovered during his training was diaphragmatic breathing and he wanted to share this technique through this app.

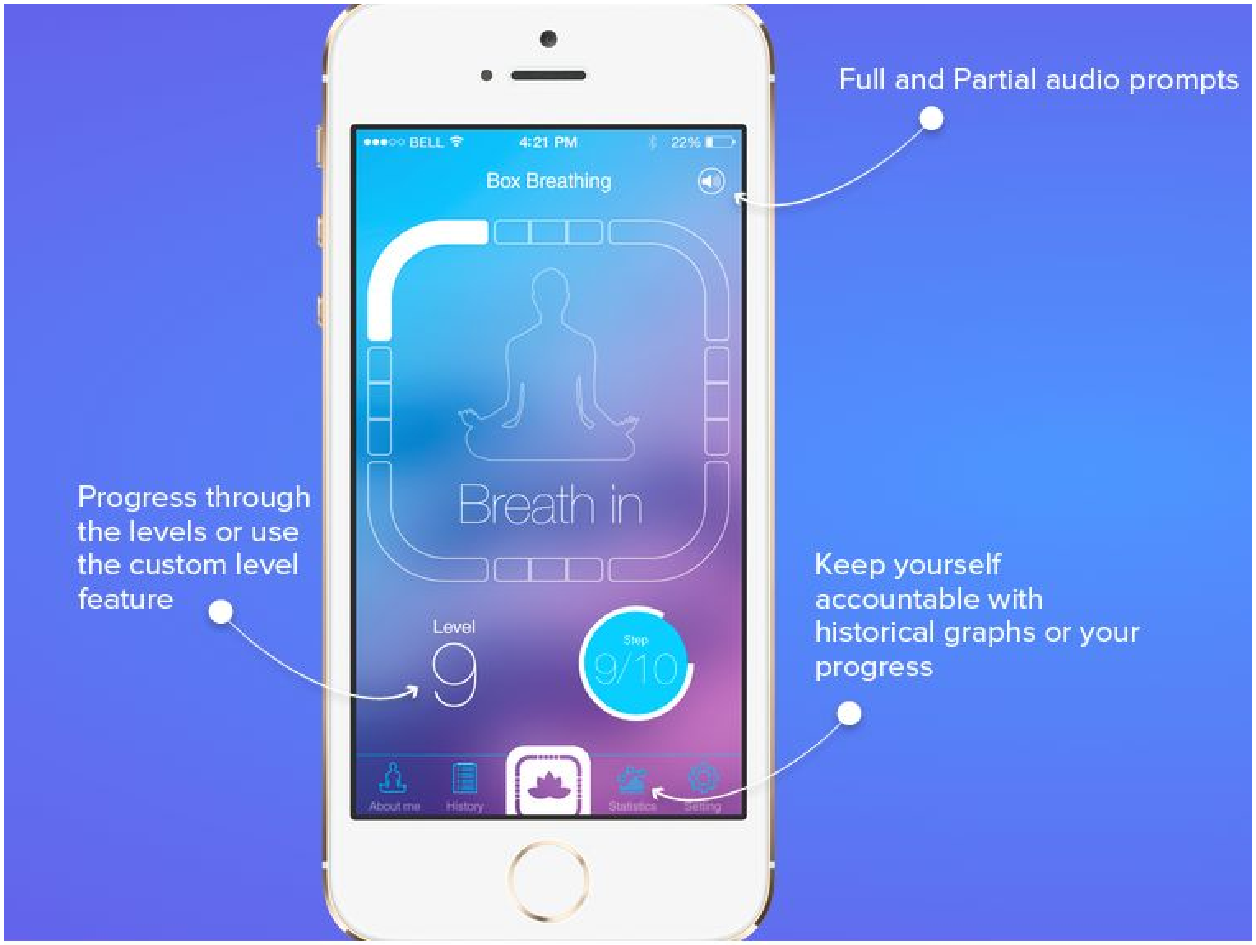

Full and Partial audio prompts
BELL
4:21 PM
22%
Box Breathing
Breath in
Progress through the levels or use the custom level feature
Level
9
Step
9/10
Keep yourself accountable with historical graphs or your progress
About me
History
Statistics
Setting

FIgure 2.14: *Breakdown of the app functionality on the homepage of BoxBreathing (Boxbreathing.org, 2018)*

In conclusion, there are no devices or services on the market that combine RIP with diaphragmatic breathing training. The devices that come closest to this concept are the Prana (Figure X) and Spire Stone (Figure X). The advantages of those devices is that they monitor breathing in real time and are therefore able to provide immediate feedback as opposed to an overview that the current iteration of the prototype used for RIP in this project is able to provide. A wearable that clips onto the waist is far less intrusive compared to RIP however the trade-off with RIP is between intrusiveness and measurement accuracy. Unfortunately there was no information on the classification of the signals on the websites of the products reviewed, however this is understandable due to the technology being proprietary and not open source.

2.4 Decisions for Ideation

RIP has been validated as an accurate way to measure tidal volume during respiration and shall be used for the data collection for this project. Diaphragmatic breathing will be taught to the participants by a trained professional without live feedback due to the scope of this project. This is also acceptable as the coach will be chosen on the basis of their experience. The training should be performed in a setting the participant is present in every day so as to cultivate the habit in response to said context/setting. Simple linear methods of classification will be applied to the data and compared to non-linear methods.

A few points of discussion are that some of the studies are slightly outdated or done in extremely narrow contexts. No studies were found covering the overall effects of diaphragmatic breathing on well-being and health. It is worth noting that a number of studies were done in South Korea, China and India reflecting a more natural approach to interventions rather than a medical one, that is in parallel with the approach of this study. Since the studies were done in Korea and China, unfortunately, only the abstracts were translated into English. This slightly weakens the analysis of those papers as all the information on the studies was obtained from their abstracts. The explanations of the relation between breathing and the results measured introduces the argument that western medicine and science may not have developed sufficiently to study complex processes such as breathing. The use of RIP was validated but only recommended under certain conditions which should be taken into account when forming the test conditions in this project. The type of classification as well as features extracted will be kept simple until proven incapable of handling the classification task i.e. using linear discriminant analysis or logistic regression. Non-linear classifiers have been shown to better deal with asynchronous breathing however the participants of this study will be trained by a professional, thus the risk of asynchronous breathing is lowered.

3. Methodology and Techniques

This chapter will introduce the reader to the various methodologies used by a Creative Technology student while undertaking a design project. Interviews, brainstorms and other such tools such as analysis frameworks are covered.

3.1 Creative Technology Design Process

This subsection focuses on familiarizing the reader with the design process behind a Creative Technology project. This design process was introduced by Mader, A. H., & Eggink, W. (2014) and a diagram of the process can be seen in Figure 3.1. The four phases of the design process from Figure 3.1 are:

- **Ideation**
- **Specification**
- **Realization**
- **Evaluation**

Ideation

The ideation phase is where a design question in the form of an idea, a client’s desire or other creative inspiration is used as a starting point (Mader, A. H., & Eggink, W. 2014). Relevant information is acquired using methods such as a Stakeholder Analysis (a process described below), interviews and brainstorms in order to create specifications that guide the realization phase. Ideas are evaluated using methods such as an iPACT analysis (People, Activities, Context, Technologies).

Specification

Prototypes are used with feedback loops and evaluations in order to explore the design space and gauge user reactions. Their reactions are used to further refine and update the specifications for the final prototype. Methods such as a FICS analysis (functions, interactions, contexts and services) or MoSCoW (Must, Should, Could, Won’t have) are used to refine, prioritize and even alter requirements.

Realization

When the updated specification list from the specification phase has been obtained, the realization phase can begin. The prototyping phase can be done in several iterations. Each prototype is developed, tested and evaluated. In this project, the prototype will constantly be tested by the user and the final version will be tested with experienced diaphragmatic breathers.

Evaluation

Functional testing and user testing are two common ways of evaluating a prototype against its specifications. The Functional testing will be conducted in Section 7, and where the program will be tested in a real world usage scenario. The testing will not involve gathering users' opinions on the device and system which are common practices with user testing.

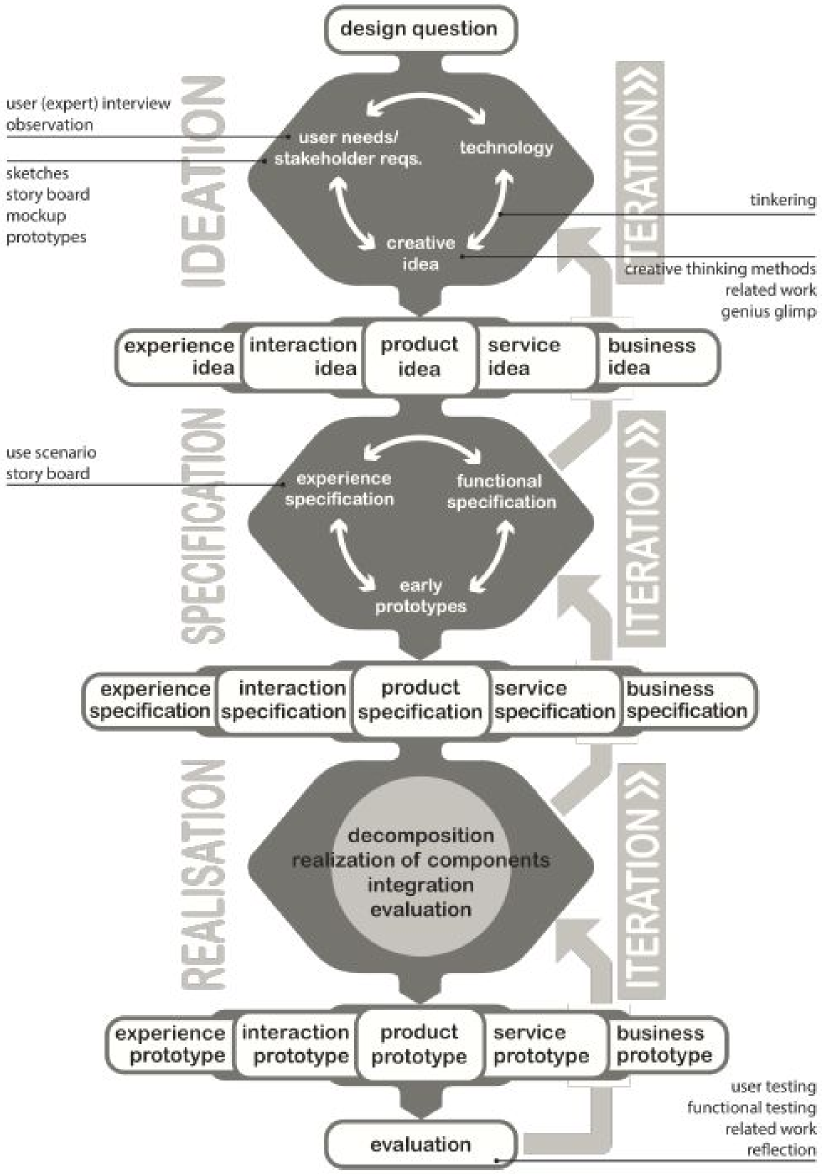


Fig 3.1: *The Design Framework for Creative Technology (Mader, A. H., & Eggink, W. 2014)*

3.2 Brainstorms

A brainstorm is an idea generating exercise where ideas revolving around central themes, topics or questions are generated in a free-flowing open discussion. Ideas generated during the activity are not criticized or analyzed until after the discussion in the spirit of idea generation and to not interrupt the flow (BusinessDictionary.com, 2018). Brainstorms can be conducted individually or in a group setting, both of which are done in this project. There are various different brainstorming styles (David, n.d):

Round robin

Participants are made aware of a central topic and are then asked to write down their ideas. Then, the participants are all given a turn to voice their ideas and opinions one at a time regarding a certain topic. Each participant is given the same amount of time, allowing equal sharing of ideas from all participants. This method is designed to generate the most possible answers to a question or solutions to a problem and doesn’t allow the ideas of one participant to influence another.

Listing

Listing is a method where an individual writes down ideas they have about the topic being brainstormed on a piece of paper. Ideas, words and phrases can all be written. The ideas should not be judged upon feasibility and a time limit can be set to spur the creation of new ideas. Like the Round robin technique, this method allows for the generation of as many answers as possible.

Free form

Participants are all made aware of the central topic, for example vacation ideas, and the participants write down their individual answers to the problem. The participants then share their responses with each other and work together to come up with the most appropriate list of ideas.

This project will combine the use of these three brainstorming methods. The participants will be informed of the current topic being brainstormed. The participants then take the time to write down their ideas using the listing method, for which a time limit is not set. Then a variation of Round robin is conducted where the participants all voice their ideas one by one after writing them down, however the other participants are allowed to add new ideas if they are inspired. Finally, after all the participants have voiced their opinions, the group performs free form brainstorming to evaluate all the ideas and select the ones that best represent the group’s opinion or solution to the topic.

3.3 Expert Review

The various ideas generated as a result of brainstorms and interviews will be reviewed with an expert, in this case the critical observer Ainara Martinez. Based on relevance, feasibility and practicality the ideas will be stricken from the list until a collection of viable, implementable ideas will be left over.

3.4 Interviews

In order to better understand the needs and desires of potential users regarding feedback for a training application or service in general, interviews and group brainstorm sessions will be organized. There are three different types of interview styles; structured, semi-structured and unstructured interviews (Rose, 1994).

Unstructured interviews: Unstructured interviews are where the interviewer does not prepare questions ahead of the meeting, and allows the conversation to flow freely, without guiding or probing the interviewee (Rose, 1994).

Semi-structured interviews: Semi-structured interviews are where the interviewer focuses on themes or topics of importance. The interviewer probes and clarifies comments made my the interviewee and uses prior knowledge to process said comments. The interviewee is given the freedom to discuss the topics in whichever way they see fit and the interviewer mainly guides the conversation (Rose, 1994).

Structured interviews: Structured interviews are where the interviewer prepares questions beforehand, and limits the conversation to only the questions. The interviewer strictly guides the conversation and nothing outside the prepared topic list is discussed.

The semi-structured interview format will be utilized for this project. This decision was made due to the possibility of additional insights coming to light, outside of the questions prepared. From previous experience, this structure has led to additional topics being discussed that the author had not foreseen, but that the interviewee found relevant to the topic being discussed.

3.5 Stakeholder Analysis

To understand a Stakeholder Analysis, the term stakeholder needs to be defined. Brugha, R., & Varvasovszky, Z. (2000) define a stakeholder as “actors who have an interest in the issue under consideration, who are affected by the issue, or who - because of their position - have or could have an active or passive influence on the decision-making and implementation process”. Sharp, H., Finkelstein, A., and Galal, G., (1999) further add to this by providing 4 categories in which to divide the stakeholders:

- ***Users:*** people interacting with the product on a daily basis (primary), on an infrequent basis (secondary) and who do not interact directly with the product but are affected by its introduction (tertiary).

- ***Developers:*** people working on and developing the product or system in question. Their interest in the final product differs from the users.

- ***Legislators:*** any group or body that provides provisional guidelines for the usage of the product or service in question that affects the development or operation of said product or service.

- ***Decision Makers:*** financial controllers and managers of the development team i.e. anybody within the development or user organization who maintains a decisive role in the development of the product or system.

The stakeholders are then ranked on “Interest in Project” and “Influence on Project” in terms of low, medium or high. This ranking is done in order to better understand the stakeholders with interest in the end result of the project and with considerable influence on the project itself.

3.6 iPACT Analysis

The iPACT analysis helps designers understand the users and usage contexts of the system they are designing. The abbreviation iPACT stands for intentions, People, Activities, Contexts and Technologies. This analysis method arose from a Human Computer Interaction methodology in order to understand the societal and technological aspects of an idea or concept (Reinius, 2011). The intention is described so that the goal of the product or system is made clear. People are modelled using personas that describe possible subsets of users. Personas describe their physical attributes, lifestyles, habits, likes and dislikes. Activities involve how the user interacts with the product; interface design, response time of the system and regularity of activities performed. Contexts can be broken down into three main contexts: social, organizational and physical environment. Technologies consists of four aspects: input, output, communication and content (Reinius, 2011), and other technologies involved in the development of the project.

3.7 Requirements Analysis

There are various ways to analyze the requirements of a system. This project will prioritize requirements using MoSCoW analysis and divide them into functional and non-functional requirements.

MoSCoW

The MoSCoW method of analysis breaks down requirements into Must Haves, Should Haves, Could Haves and Won't Haves. Must Haves are requirements that the prototype needs to have without which it could not be delivered by a certain date without being safe or possibly legal. Should Haves are requirements that it would be convenient for the prototype to have but are not important for the basic functioning of the prototype. Could Haves are requirements that are even lower on the priority list than the Should Haves and there would not be a noticeable effect were they to be left out. Finally the Won't Haves are requirements that are too far-fetched or unrealistic for the decided upon timeframe but is good content for future work (Agile Business Consortium, 2018).

Functional and Non-Functional requirements

Requirements can be divided into the categories of functional and non-functional (Sqa.org.uk, 2007). Functional requirements are those that deal with specifics of what a system should do. For example, a functional requirement would be that the system must encrypt user passwords. A non-functional requirement is can be a constraint or criteria. For example, the user interface should be designed with an appealing color scheme. Non-functional requirements are usually put at a lower priority than the functional requirements.

3.8 FICS Analysis

The FICS analysis is used to visualize a usage scenario from the side of a system instead of a user like in the iPACT analysis (Larburu et al., 2013). The abbreviation FICs stands for Functions and events, Interaction and usability, Content and structure, and Style and aesthetics. Functions and events are what the system does and how it reacts to events, Interaction and usability, shows how the user is supported in carrying out an activity. Content and structure relates to how the system stores data and how it is accessed and finally Style and aesthetics describes the look and feel of the system. The results from FICS and iPACT comes together to give a better understanding of the system, its role and usage scenarios.

3.9 System Architecture

System architecture is a model of the system that gives an overview of its various components and how these components are linked together. The iPACT and FICS analyses help model the requirements for the system from which the functionality can be determined. The system architecture can have multiple levels, where a deeper level would go into the workings of an individual component of the system. Flow charts and diagrams are used to model system architecture. The levels of the architecture are as follows: the first level is inputs and outputs of the system, the next level shows the various functionalities of the system represented by connected blocks and finally the third level will show the inner-workings of each of the blocks in the previous level. The arrows connecting the blocks indicate which components are connected and the order in which they are called upon.

3.10 Cognitive Walkthrough

“A Cognitive Walkthrough is a usability walkthrough technique that focuses primarily on the ease of learning a product.” (Wilson, 2014). The cognitive walkthrough will be used to guide the reader through the data processing in the system to provide them with a better understanding of how the algorithm functions. The walkthrough will use the help of an Activity diagram to help identify the major processes, how they are related and in what order they are performed.

3.11 Activity Diagram

An activity diagram is an overview of a usage scenario where activities are split between user and the system. The diagram shows the connection between activities and the triggers of certain events. A black circle indicates the beginning of a usage scenario, a diamond shape indicates a decision being made either by the user or the system (checking whether a condition has been met), a rectangle indicates an action or process occurring and finally a filled circle within another circle signifies the end of the scenario.

3.12 Evaluation

The program will be evaluated using four methods, a Functional Test, Code Tests, Expert Validations and Program Tests. A Functional evaluation is simply a checklist covering the various requirements specified in the Specification phase and checking to see whether they are accomplished. During Code Tests, components of the program are tested on various possible scenarios to ensure the program can handle them. This goes hand in hand with the functional evaluation. For example, if a functional requirement is for the code to separate the data by the minute, then the code tests should make sure that the program can identify all the minutes in the code and segment the data by the minute accordingly. An expert validation involves approaching an expert in the field to comment on the project and methodology used. Finally, the program test, pits the program against a real world situation and checks whether it can find scheduled DBMs in a set of recorded data.

User evaluations are performed to validate and check if non-functional requirements have been met, however due to the technical scope of the requirements, this is being conducted using code tests.

4. Ideation

This chapter will answer the research sub-question:

“*What kinds of breathing features would users like to be presented for the purpose of feedback?*”

The ideation phase will explore the user requirements and wishes through the use of a Stakeholder Analysis to identify influential and important individuals. An individual brainstorm will then be conducted in order to generate ideas for the experimental protocol, classification and feedback regarding the features presented to users. A group brainstorm will be conducted with a group of potential users to brainstorm ideas for features they would like to receive for feedback on their breathing. Interviews will then be conducted involving a sample of the previously identified stakeholders. An Expert Review will be conducted based on the ideas generated from the brainstorms and interviews. An iPACT analysis will be performed in order to better understand user requirements from the system. Finally, a list of preliminary requirements will be drafted for the Specification phase.

The breathing training wearable device will be referred to as Airleviate from here on. The term “diaphragmatic breathing moment” will be shortened to “DBM” and used as such from here on.

4.1 Stakeholder Analysis

Users

A potential user of the Airleviate service can essentially be anyone wishing to improve their breathing and experience the physiological benefits of diaphragmatic breathing. Young children may use it to ensure a varied breathing routine. Students and adults may use this service to reduce anxiety caused by their studies or work respectively. Athletes can prevent oxidative stress using this service to train diaphragmatic breathing. Physiotherapists and specialists may include it as part of a treatment against dysfunctional breathing, and anxiety, in which case both the patients and then specialists are users; the latter being an indirect user. It can also be used to train diaphragmatic breathing for the purpose of martial arts, yoga or meditation; thus novices in those disciplines are potential users too.

Developers

This system has three developers: Ben Bulsink who is the main developer and creator of this service, and Florian Naumilkat who is working on the user interface that provides feedback to the user. The last developer is the author who will be developing the program used to process, classify and analyze the collected data.

Legislators

Farmatec is an governmental organization that oversees the registration of medical devices to be sold on the market in the Netherlands (Business.gov.nl, 2018). The prototype and service would need to be registered with them in order to be allowed to be sold on the Dutch market. Other legislators include groups such as lawyers and medical policy advisors. These two groups go hand in hand as policy makers themselves can be lawyers or work closely with lawyers in order to establish laws restricting or permitting the use of the service.

Decision Makers

The client of this project, Ben Bulsink is a decision maker as his goal and vision will guide this project. The supervisor and critical observer, Erik Faber and Ainara Martinez Garde are also decision makers in the project as they are overseeing its progress and providing feedback and suggestions on improvements where necessary. They also set the time frame for the project. Florian Naumilkat, a fellow student studying Creative Technology is a decision maker, as features from the data will need to be extracted and provided to him for the interface, the details of which he will worked out during the ideation phase of this project. Finally the author of this report, Arnav Mundkur, is a decision maker in terms of which classification model to employ and how to design and code the script.

Table 4.1 represents the stakeholders sorted by interest, category and influence. A graphical representation of the influence versus interest relationship is visualized below in Image 4.1. The stakeholders in the top right corner of the diagram are the most influential and interested in the project.

Stakeholder	**Category**	**Interest**	**Influence**
Children	User	Low	Low
Students	User	Medium	Low
Working adults	User	Medium	Low
Patients	User	Medium	Low
Specialists	User	Medium	Low
Medical Policy Advisors	User/Decision Maker	Medium	High
Lawyers	User/Decision Maker	Medium	High
Farmatec	Decision Maker	Medium	Medium
Arnav Mundkur	Developer/Decision Maker	High	High

Florian Naumilkat	Developer/Decision Maker	High	Medium
Ainara Garde Martinez	Decision Maker	High	High
Erik Faber	Decision Maker	High	High
Ben Bulsink	Decision Maker/Developer	High	High

Table 4.1: Stakeholder influence and interest

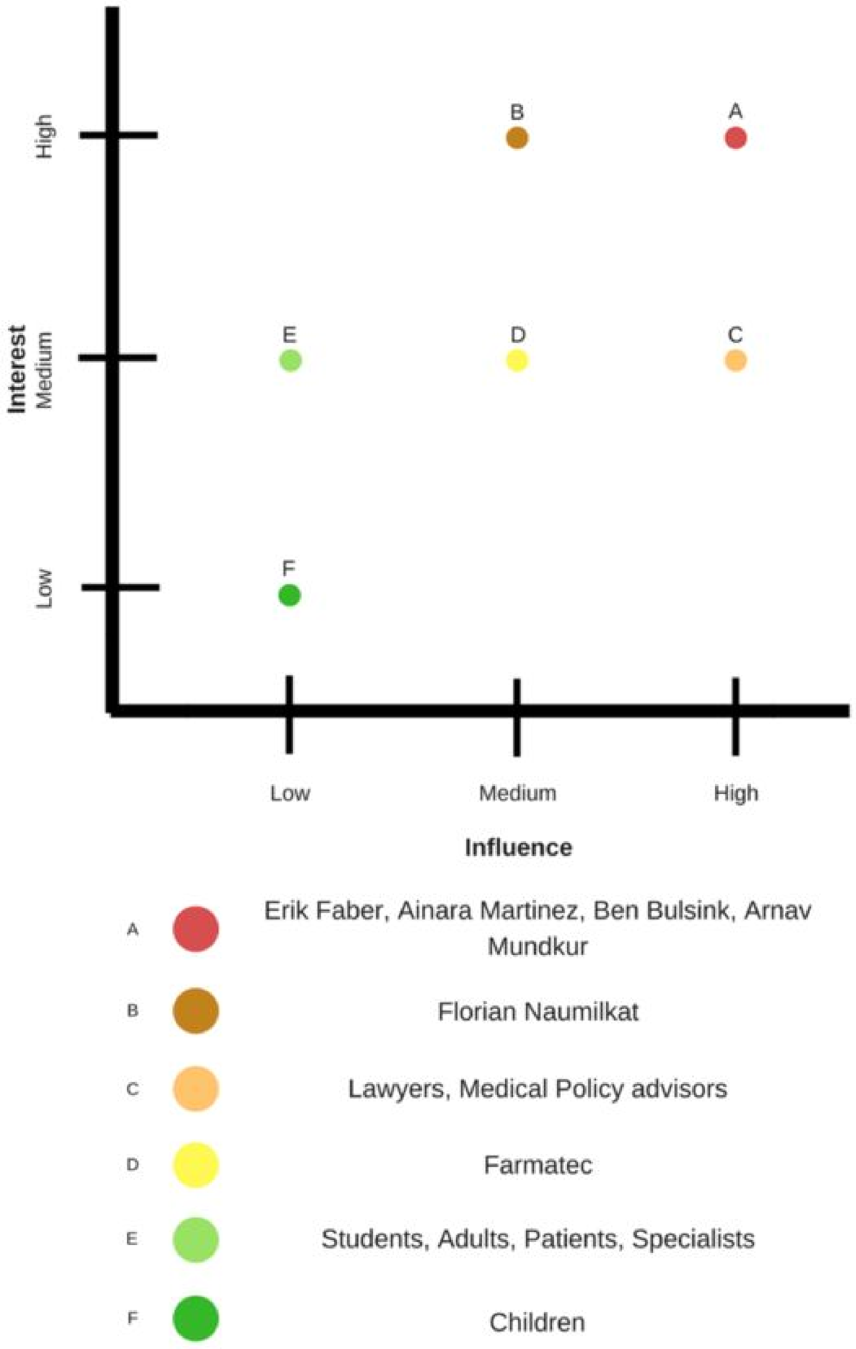


Image 4.1: *Stakeholder analysis*

To sum up the findings of the stakeholder analysis, the author, Florian Naumilkat, supervisor and critical observer Erik Faber and Ainara Martinez as well as client Ben Bulsink are the most important stakeholders. Although lawyers and medical policy advisors are important stakeholders, due to the scope of this project, they will not be approached. Students and adults, in the form of teachers, are easily available and will therefore be interviewed, to understand the needs of a potential user.

4.2 Individual Brainstorm

This project has four major aspects, classification of RIP and accelerometer data, analysis of RIP data, measuring respiration and providing feedback to a user. An individual brainstorm was conducted by the author in order to generate ideas for classification, methods to analyze the data, a protocol for recording breathing data and feedback to present a user. A mind-map was used to document ideas in an appropriate way and can be found in Appendix 11.B.

Experimental protocol

Participants will be recruited and introduced to the project and its goals, and the experimental procedure will be explained to them. At the beginning of the recording session, the participants will be instructed to perform 2 and a half minutes of chest breathing and 2 and a half minutes of diaphragmatic breathing in the seated and standing positions, totalling 10 minutes of breathing data. The accelerometer data that was recorded during those 10 minutes will be compiled to train the classifier on the "Stationary" category of movement. Then the participant will be requested to walk around the room for 10 minutes to record "Moving" data with the accelerometer for the movement classifier to be trained on. Following this, the data will be compiled into text files for the Chest breathing data, Diaphragmatic breathing data, Stationary and Moving activities. The participant will then be asked to write down three times during the next hour. Timers are set for the three times listed. The participant then goes about their own routine for an hour while wearing the Airleviate to record their breathing. When the timers go off, the participants are asked to focus on breathing diaphragmatically for a period of three minutes. After each three minute period ends, they continue going about their routine.

Classification

The training dataset is the data that will be used to train the classifiers used to identify periods of diaphragmatic breathing. As for the classification itself, the prediction accuracy of a linear classifier will be compared to the prediction accuracy of non-linear classifiers. The classifiers to test can be Logistic Regression, Neural Networks, and Support Vector Machines discussed in Section 2.13. A factor to keep in mind is the low ratio of training data to unlabelled experimental data. The user should be able to specify a threshold signifying how tolerant the program is with the performance of their diaphragmatic breathing, or in other words how strictly it punishes accidental chest breaths during a DBM. The higher the threshold, the stricter the classification to qualify as a DBM will be. A "restful" moment is considered to be a period of time where the user is either sitting, or standing still. The accelerometer data will be classified in a similar fashion to breathing, where the accuracies of various models is compared and the most accurate is chosen.

Feedback

The following features for feedback were a result of the brainstorm:

- The number of "restful" moment.
- The average breathing frequency per minute.
- The average contribution from the abdomen during DBMs and the times at which these moments occurred.
- The total time spent breathing diaphragmatically.
- A quality factor, representing the diaphragmatic breathing performance during a recorded breathing session.

For the rest of the features, please refer to the mindmap in Appendix 11.B.

Analysis

The RIP signals can be analyzed in the time and frequency domains. In the time domain, the envelope of the signal can be used for analysis as well as counters to count certain events like DBMs and repetitive breathing. Statistics can also be used for example the average breathing frequency or the average duration of DBMs for that session. In the frequency domain, the frequencies of the signals can be analyzed to measure the breathing frequency, for example by using Fast Fourier Transform (FFT). This can be used to help identify frequencies of interest. Please refer to the Appendix 11.B. for the mindmap. The data will be segmented per minute, so that all the samples belonging to a certain minute are grouped together.

4.3 Group brainstorm

A group brainstorm was conducted in collaboration with Florian Naumilkat, the Creative Technology student working on the design of the user interface. The group consisted of 6 participants that were fellow students from the third year of Creative Technology, who were either interested in the project, or had volunteered to be part of the data collection process. The group brainstorm touched upon the themes of data visualization and the data or statistics that they would want to see regarding their breathing. The ideas regarding visualization and features are presented below:

Data visualization

All the participants mentioned in some form that they were unaware of the benefits of diaphragmatic breathing and therefore would find it useful if the interface could provide useful facts about breathing and the benefits of diaphragmatic breathing based on scientific literature. The participants all had various creative ideas for displaying and visualizing the data. For the complete description of the ideas discussed on Data visualization, please refer to Florian Naumilkat's report.

Features

During the brainstorm, the participants including the "experienced" diaphragmatic breathers, found it difficult to name statistics that they would like to know about their breathing performance off the top of their head. For the ideas relating to the growing of a garden, or energy for a game, a statistic such as a quality factor or total time spent breathing diaphragmatically could be used. During the discussion, the participants mentioned features such as:

- The total time spent breathing diaphragmatically during the day, as well as time intervals during the day that they could zoom in on depending on their interest (group time in segments of 5 or 10 minutes).
- Breathing frequency.
- Time-stamps relating to the periods of diaphragmatic breathing for backtracking.
- The percentage of time during the recording period where diaphragmatic breathing was used.
- Statistics could be compared to the average user, sorted by gender and age, to introduce a slight competitive edge.

4.4 Stakeholder Interviews

From experience with interviews during the Background Research phase, the semi-structured format of interviews are the most practical. This is because the questions formulated before the interview are usually uninformed and are subject to be updated or scratched during the interview itself. The free flowing nature of semi-structured interviews allows the interviewee to make comments that would otherwise be lost in the rigidity of a structured interview. It is also the experience of the author that interviewees often reveal new themes or angles previously ignored or not thought off when preparing questions or topics of discussion.

There will be three interviews conducted; one with the client Ben Bulsink, to understand his goals and what he would like to see in terms of classification and feedback. The next interview will be with a 20 year old student from the second year of Advanced Technology, which is a subset of the potential stakeholders. Only one student is approached as 6 students were approached in the group brainstorm, and this student has noteable experience with wearable devices. The final interview is with a teacher who has been using the Spire Stone breathing device described in Section 2.3.1.1, in order to learn about her experiences with another breathing training device. The questions to these interviews can be found in Appendix 11.C.

4.4.1 Interview Ben Bulsink

- Mr. Bulsink was inspired to work on and develop the Airleviate after a discussion with Parviz Sassanian about breathing, who is an acupuncturist and eastern medicine practitioner interviewed in Section 2.2.1. Mr. Sassanian speculated that the Airleviate could potentially help some of his patients and a larger population of users.
- His personal experience with breathing awareness during his choir influenced his interest on the Airleviate too. Breathing and awareness allowed him to calm himself and ground himself in the moment.
- He can see the system being recommended to patients under breathing experts and others in the medical field.
- How often a user has “restful” moments was a statistic of interest to Mr. Bulsink. From his own experience, he takes a short pause between each exhalation and inhalation of the next breath and was curious to see if other people breathe in the same manner.
- Mr. Bulsink was interested in moments of “periodic breathing” where the breathing remains fairly constant.
- The medical domain is the main area where he sees this technology being applied. Especially in the case of dysfunctional breathing. Wearing the Airleviate, due to its intrusiveness may nudge and therefore train a certain type of behaviour. The reaction to breathing would be more mental rather than a habitual state.
- Haptic feedback definitely has its place in future iterations, once real time feedback is realized. There was indecision on whether to use the carrot or stick feedback. From his experience and from the people he had spoken to, encouraging feedback (carrot) has shown more results than negative feedback (stick).
- Mr. Bulsink does not see himself adopting the technology once it hits the market and is unsure who else will adopt it and for what reason.
- A notification system he thought would be ideal rather than the haptic feedback, would be patterns on a graphical device, abstracts from the data represented. However he expressed staying away from short loop feedback, because then the mind is always in between and processing the feedback instead of focusing on the activity itself.

4.4.2 Interview with Student

- Loves recording bio-data and tries to wear his Apple watch at all times to record his activity. He is really fond of the Apple Health and Activity apps that display the statistics of his physical performance for a day, such as the total distance walked or total flights of stairs climbed (Apple A, n.d; Apple B, n.d.).
- The student also uses training and workout apps like Freeletics and Runtastic (Freeletics.com, n.d; Runtastic.com, n.d.). He is fond of the the gamification elements of providing a score with every work-out based on the time it was completed in. He appreciates the diversity of exercises offered in the workout apps and the choice of combinations of exercises.
- He finds that the fitness apps that are available today, those linked to exercise, are lacking the holistic aspect of involving diet and sleep. He was enthusiastic about an app that combined different aspects of life and didn't just isolate it with respect to one activity.
- The student likes wearable devices very much and not only has an Apple smartwatch, but Under Armor running shoes with built in accelerometers.
- He said he doesn't pay much attention to his breathing outside activities such as exercise, but has realized that he has periods of shallow breathing followed by deep breaths when he studies. He is aware of diaphragmatic breathing and has practiced Yoga.
- The student stated that he wouldn't like smartphone notifications regarding his breathing if the data was recorded in real time, however he did mention that haptic feedback implemented on the wearable itself or linked to his Apple smartwatch would be ideal.
- With regards to statistics of his activities, he expressed his liking of solid figures such as the total time he spent standing during the day, the number of flights of stairs he climbed or the total distance he walked that day.

4.4.3 Interview with Teacher

- Textual and statistical feedback was too commonplace among many training apps nowadays and the essential aspect of haptic feedback (physical feedback) is missing. Something as physical as breathing requires physical feedback. In addition, there is too much cognitive load from receiving statistics and data in an accompanying interface. Real time feedback is useful for tying the feedback to mistakes in order to facilitate proper learning.
- The FitBit was another wearable worn by the interviewee (Fitbit.com, n.d.). There were a few complaints that they had regarding the usability in terms of battery life, screen viewability in sunlight and the inaccuracy in measurements and classification. The last factor was the biggest contributor to the interviewee abandoning the device.
- The interviewee could not give an example of an interface they enjoyed using in combination with a wearable device, and regarded them as quirky. The heart of what they looked for was in the real time feedback.
- The interviewee's own interest in the the monitoring of their daily activities and the associated biodata spurred them to try out various wearables such as the Spire Stone. The interviewee is

also a researcher into wearable technology at the University of Twente and therefore also has an academic interest in wearable technology and wearable activity tracking devices.

- A complaint the interviewee had with existing technologies is that some of them did not have the capability of being integrated into other platforms. The example given was data acquired from the Spire Stone could not be used in combination with a workout app. The FitBit on the other hand did provide options for integrations with other services and apps.
- Some downsides of the haptic feedback experience the interviewee had was that the Spire Stone sent complicated haptic feedback and they had to decoded all the buzzes which became tiresome and distracting. A simpler version of it would have sufficed and actually have been better received. Haptic feedback technology also suffers from the fact that it isn't very private and the buzzes can be heard by other people in the user's immediate vicinity.
- With regards to instructional feedback, haptic feedback used for reminders is a well appreciated concept, however the user needs to be introduced to the idea and prepared to receive it.
- In particular a breathing statistic they would wish to see is the number of and duration of periods where the user is calm and or relaxed. This can help the user reflect on the activities they were doing during those periods, in order to better understand what calms them.
- Activities where a device that reminds you to breathe would be useful are activities that require large amounts of concentration. The example given was a yoga class, where while attempting to hold a certain pose, the user forgot to maintain their breathing. Haptic feedback would prove to be very useful in such situations.

4.5 Expert Review

The author met with Mrs. Ainara Martinez Garde to discuss the approach to dealing with “periodic” breathing and diaphragmatic breathing classification. The data analysis plan to segment the filtered data by the minute, analyze it based on breathing frequency using peak detection and peak to peak difference, was explained to Mrs. Garde. She recommended comparing it to the results of an FFT analysis as this can be used to confirm the peak difference. Features that could be used to identify the breathing frequency in an FFT graph would be the kurtosis (steepness) of the peak, the magnitude and the area under the peak within a certain frequency range. However, she mentioned that in her work, she rarely used FFT and preferred Spectral Power Density Analysis using a periodogram. This uses the power of the signal at certain frequencies instead of the amplitude.

Mrs. Garde was a little confused as to the usage of the term “periodic” breathing and after explaining that Mr. Bulsink was interested in moments of continuous or similar breathing rates, she recommended “breathing similarity” or “repetitive breathing” would be a better term for it; as she explained that periodic breathing is a breathing condition patients suffering from heart failure experience. It consists of periods of hyperventilation followed by periods of hypoventilation, and is in fact not what this project is interested in. In order to classify a repetitive breathing moment, Mrs. Garde recommended using the breathing frequency per minute and seeing how it deviates from the breathing frequencies of the previous two minutes. From now on, the term “Repetitive breathing” will be used instead of “Periodic breathing”.

The author explained that the classifiers were currently being trained on the filtered signal data from the two bands and Mrs. Garde expressed the importance of providing features along with the data to train a classifier better. In deep learning, a classifier is provided with a large volume of data from which if calculates various features, and filters the important ones to then train itself to classify something. This is not the case in machine learning, where features should be included in the training dataset. Features that she recommended were basic statistical ones such as the mean, median, standard deviation and interquartile range. Another statistic that she recommended was the amplitude after normalization, however a more accurate feature would be the area under the curve of a peak in the signal (the area under the inhalation and exhalation part of the signal.

Mrs. Garde also highly recommended that the data be normalized, so that the data recorded from different measuring devices does not affect the classification. The data can be normalized by subtracting the mean and dividing by the standard deviation of the data. She also said that the low pass filter with a cut-off frequency of 0.7 Hz does restrict the data somewhat and doesn’t allow for the recording of breathing during activities with heightened heart rate like exercise, however since that is out of the scope of the project she said this was agreeable.

4.6 Preliminary Concept

Based on the findings from the stakeholder interviews, brainstorms and expert interviews, the following ideas come together to form the preliminary concept for collecting, processing, classifying and extracting features from the data.

Features

It was found that periods of diaphragmatic breathing and periodic breathing are of interest to users. Features such as the longest DBM, timestamps of each DBM, a quality factor representative of the performance during the session, and the contribution of the abdomen during DBMs were of interest. Similar statistics to these like the average DBM length, the number of DBMs will also be included.

Data Processing

The breathing frequency will be calculated using a periodogram instead of FFT as recommended by Mrs. Garde in Section 4.5. The breathing data should be normalized to avoid data recorded using different devices affecting the classifier prediction accuracy. The frequency range outside which frequencies should be filtered is 0.2 - 0.7 Hz.

Classification

The classifiers should be trained on more than one set of features as it should yield a higher prediction accuracy. The prediction accuracies of various classifiers will be tested to find which is the best at classifying RIP and accelerometer data in particular. Data will be grouped and classified by the minute. “Periodic” breathing will be replaced with “Repetitive” breathing according to the definition provided by Mrs. Garde. Classifiers will be classifying individual samples not only based on the raw data values but others such as the respective contribution values to the sum of the two bands. Given that repetitive breathing can be calculated based on the breathing frequency and compared to the breathing frequency of previous minutes, the user should be able to specify how many minutes they want a minute to be compared to.

Experimental Protocol

The participants will be asked to record training data for the classifier by performing diaphragmatic breathing in the seated and standing positions. Participants will also be asked to record data as they walk around the room in order to get accelerometer data to train the movement classifier. The participants will then record their breathing for around an hour to identify DBMs.

4.7 iPACT Analysis

An iPACT analysis will be conducted to better picture the potential users of the system and their interactions with it. For an explanation of this methodology, refer to Section 3.6.

Intention

The intention of this system is to qualify and quantify breathing data through classification and analysis to provide users with feedback in order to help them cultivate the habit of diaphragmatic breathing and to inform them of their breathing performance throughout the day.

People

Daphne is a 25 year old student doing her masters in Embedded Systems at the University of Twente. She spends long afternoons and some of her weekends at the library, pouring over previous lectures. She suffers from fear of failure due to her intense study, and does not participate in any extracurricular activities. Daphne has tried using meditative apps in order to help her maintain her calm but is not having much success with them. She is not only interested in tracking and training apps, but the biodata processed by them. She is unsatisfied with the existing apps available as they do not allow for viewing of the raw data.

Rick is a 34 year old Science teacher at the Rivers International School in Arnhem. He teaches Science to the 2nd and 3rd grade students. His students are rarely interested in his material and are usually playing pranks on each other or staring out of the window. Rick has developed a short temper due to this and regularly loses his patience with his students. When Rick isn't correcting essays or exams, his enthusiasm for technology manifests itself as he pours over articles on the verge.com about the latest in wearable technology. He loves tracking his movement and never goes anywhere without his Spire Stone. Rick spends his time experimenting with new wearable technology and tracking his activity and recording biodata. He uses his smartphone to download accompanying apps to view statistics of his activities after a long day.

Activities

Airleviate can be worn all day or during specific activities where the user wishes to have their breathing monitored. Breathing rates for adults typically lie between the 0.08-0.7 Hz range (Fleming and Tarassenko, 2007). Every user will have a fixed position that they will place the abdominal and thoracic bands at. The user can opt to wear the bands over an undershirt or on top of their regular clothes. At the end of the day, the user will get an overview of their breathing activity through an interface either on an app or website. The user removes the bands and charges the Airleviate for its next use. The interface shows the user statistics from their day's breathing and compares their performance, using the quality factor, to the inputted goal quality factor that the user set when they first started using the app. The user can also adjust the threshold level which determines how strictly chest breaths are taken into account when classifying diaphragmatic breathing for a minute of data. Likewise, length of the buffer can be adjusted, which is used to compare the average breathing frequency of a minute to identify repetitive breathing moments.

Context

The Airleviate has been designed to be worn all day, in any social context. In its current iteration, the feedback can only be viewed at the end of the day. Examples of contexts the Airleviate can be worn in are while a student studies at the library, a teacher wearing it while giving a lecture and a yoga student wearing it while performing their routines. There are no organizational contexts, however a physical environment where the Airleviate can be worn is at a yoga center or Tai Chi class.

Technology

General

The Airleviate uses Respiratory Inductance Plethysmography for the data acquisition, a microcontroller (Arduino) is used to store the data. The Arduino uses serial communication over a USB cable to transfer the data to a desktop application that outputs the data as a text file (the input). The contents of the text file are the X, Y and Z axis data from the accelerometer and the readings from the two bands. A python script then loads the text file, processes the data and uses the library Scikit-learn to call functions that help classify the data. Statistical analysis is performed on the data in order to extract features such as breathing frequency that are then sent to the interface application, as the output of the program.

Programming languages and libraries will be investigated to find the best fit for this project. Matlab and Python are a program and programming language respectively, that the author is familiar with.

Matlab

Matlab is a programming platform used by engineers in the field and students alike and can be used to model physical phenomena, develop algorithms and analyze data (Mathworks.com, 2018). The author has used Matlab in the past for basic classification assignments involve accelerometer data. Matlab uses its own proprietary language for scripting and there are many libraries and functions developed for the purpose of machine learning, deep learning, signal processing and image processing (Mathworks.com, 2018).

Python

Python is an object oriented programming language that allows for manipulation of high-level data structures and is a popular tool for rapid prototyping and development (Python.org, 2018). Python has extensive libraries ranging from web development (Django) to Machine learning (TensorFlow). TensorFlow is an open source framework used for “high performance numerical computation”, and can be used to generate Neural Networks for the purpose of classification (TensorFlow, 2018). Scikit-learn is another open source machine learning library for python (Scikit-learn.org, 2018). Scikit-learn can be used for classification and supports a large number of classifiers including support vector machines, logistic regression analysis, and supervised neural networks (linear and non-linear classifiers). Python also has libraries for signal processing and image processing and is a popular tool among data scientists.

Since Scikit-learn supports both linear and non-linear classifiers, has a more user-friendly development environment and has extensive tutorials and online documentation, it shall be used for this project.

Classification algorithms

The two sets of data that need to be classified are RIP respiratory data and accelerometer data. In literature, various algorithms have been used to classify accelerometer data. Prasertsung & Horanont, (2016) used a Support Vector Machine (SVM) to classify tri-axial accelerometer data, whereas Kuspa and Pratkanis, (2013) employed Gaussian Discriminant analysis. Ravi et al. (2005) used a Decision Tree with which they achieved a classification accuracy of 84%. This is slightly lower than the accuracy achieved by Kuspa and Pratkanis which was 92%. However, some signal preprocessing was involved. Data will be recorded using the accelerometer for the activities of sitting and standing, and various classifiers will be trained to compare their prediction accuracies. .

As for the breathing data, considering the Preliminary Concept in 4.6 and following the findings of the background research on classifiers; each classifier is unique and has its own strengths. Therefore, breathing data will be classified using the linear classification algorithm, logistic regression and using non-linear classification algorithms such as Neural Networks and Support Vector Machine. The accuracy of the models will then be compared and the most accurate model will be chosen.

Usage scenario

Daphne has an upcoming exam for her course on Embedded System Architecture that she has been struggling. Daphne wakes up early in the morning and while getting dressed, places the abdominal band around her navel and thoracic band over her chest and under her armpits. She has a quick breakfast and heads straight to the library. She has to get through four lectures today and two of them involve a lot of calculus that she has always struggled with.

Over the course of the day, Daphne gets restless as the answers she arrives at from the tutorial questions and the answers provided do not match up at all. This aggravates Daphne and she gets up to get a coffee for a break. As she moves from her seated positions she becomes aware of the abdominal band and reminds herself to take some deep breaths to calm herself down. She remains seated and uses the diaphragmatic breathing technique she read about in the instructional manual of the wearable she was recommended by a friend to help her deal with her frustration and anxiety. After 5 minutes of doing so, she opens her eyes and decides against the cup of coffee and instead persists with the problem until she solves it correctly.

Daphne leaves the library around seven in the evening and cycles home via the shopping center in order to get herself some dinner. She saw the last of her favourite salads being taken by the person who got to the cooler five seconds before her and she gets frustrated. She clenches her body as an inward expression of her frustration and is interrupted by the feeling of the abdominal band loosen slightly. Daphne closes her eyes and reminds herself to breathe deeply and after a few seconds, she composes herself and decides to buy a pizza from the frozen section.

When Daphne arrives home and puts the pizza in the oven, she is on her phone when she thinks of her breathing and removes the Airleviate and connects it to her computer. The program takes a few minutes to run and the statistics from the day are displayed to her. She sees that she had three restful moments, two of which she could identify from the incident at the library and at the shop. She sees that she had a few diaphragmatic breathing moments during the afternoon when she was in the library, however her quality factor was lower than yesterday's value, so she reminds herself to breathe more diaphragmatically the following day. She sees that her breathing is very repetitive when she is working and is reminded of how she holds her breath when she reads long paragraphs of text.

4.8 Preliminary Requirements

Following the findings from the Stakeholder analysis, iPACT analysis, individual brainstorm and interviews with various stakeholders, a list of preliminary requirements is determined and listed below. These requirements may be subject to change during the Specification and Realization phase.

1. Use multiple breathing values from RIP data to classify the data as “Diaphragmatic” or “Chest” breathing.
2. Data classified as “Moving” or “Stationary” for the restful moments using accelerometer data.
3. Classify breathing as “Repetitive” or “Non-Repetitive” to describe repetitive breathing using the breathing frequency obtained from a periodogram.
4. Filter breathing data to exclude everything outside the 0.2-0.7 Hz range.
5. Segment data by the minute by using a sliding window of 1 minute.
6. Export the data as a text file to send to the interface.
7. User can enter the threshold and buffer length values when the program starts up.
8. Include safeguards to ensure proper parameters entered to run the script.
9. Use multiple means of classification and compare the results of each to find the optimal one.
10. Include the following features in the feedback:
    a. Total recording time of the session.
    b. Calculate a quality factor based off diaphragmatic breathing throughout the recorded session.
    c. Total time spent breathing diaphragmatically.
    d. Count the number of diaphragmatic breathing periods.
    e. The longest DBM.
    f. Times of the longest DBM.
    g. The average length of a DBM for that session.
    h. Provide time stamps associated with diaphragmatic breathing moments (DBM) to allow users to back-track and reflect on their activities.
    i. Include the average contribution of the diaphragm during DBMs.
    j. The average breathing frequency per minute for the feedback.

    k. The longest periodic breathing moment.
    l. The times of the longest periodic breathing moment.
    m. Calculate total duration of time spent breathing periodically.
    n. List the repetitive breathing moments along with their times and the duration of the moment.
    o. The number of periodic breathing periods during the recording session.

    p. The longest restful moment.
    q. The times of the longest restful moment.

r. Calculate total duration of time spent stationary.
s. List the restful moments along with their times and the duration of the moment.
t. The number of restful moments during the recording session

11. Export the data in JSON format.
12. Airleviate should support haptic feedback.
13. Airleviate should use haptic feedback as reminder and or notification system.

Certain requirements are explained in further detail below:

Requirement 2 states that movement will be classified as “stationary” or “moving”. While recording the breathing data, participants will be standing or sitting still. The accelerometer data is compiled into a file for stationary data, and likewise the data recorded while the participant walked around the room is compiled into a file for movement data. The movement classifier will be trained to distinguish between a user moving, and sitting or standing still.

Requirement 3 regarding classification of repetitive breathing as Mrs. Garde suggested, will be determined by whether the breathing frequency of a minute falls within one standard deviation of the breathing frequencies of the minutes in the buffer.

The quality factor mentioned in Requirement 10b will be calculated by dividing the total duration of time spent breathing diaphragmatically by the total duration of time of the recording. This is the single descriptive statistic of the user’s overall diaphragmatic breathing performance.

Requirement 10e mentions the longest DBM which will be found by finding the start and end times of each DBM, and the DBM with the longest duration is used for this statistic.

Requirement 10i discusses including the contribution of the diaphragm, where the values of abdominal band and chest band are summed and the contribution of the abdominal band to said sum is reported along with the start and end times of a DBM. This is to show the user how much of their diaphragm they were using while breathing diaphragmatically.

Requirements 10 k-o and p-t are similar statistics to the ones for diaphragmatic breathing, but applied to restfulness in relation to movement, and repetitive breathing.

4.9 Conclusion

Individual and group brainstorms as well as stakeholder interviews were used to answer the research sub-question:

"*What kinds of breathing features would users like to be presented for the purpose of feedback?*"

From the brainstorms, it was found that users are interested in statistics like a quality factor to describe their performance during a session, the longest DBM and the average DBM duration. Similar features were thought up during the in individual brainstorm, showing that the author and potential users have similar ideas in mind for what the breathing feedback should consist of. The interviews showed that the client had an interest in moments of rest (where the user is relatively stationary) and moments of periodic breathing (where the user's breathing frequency does not vary too much with time). The complete list of features can be found above in the Preliminary Requirements section.

5. Specification

In this chapter, the preliminary requirements listed in Section 4.8 will be analyzed. A FICs analysis will be conducted to analyze the usage of a system from the perspective of the system. A cognitive walkthrough will then be provided to guide the user through the processing of the data. This is followed by an Activity Diagram in order to show a usage scenario between system and user and the interactions between the two. An overview of the system architecture is provided to give the reader an understanding of the inner workings of the program and the functions required for the data processing. Finally, the chapter ends with a prioritization of the requirements that will be used in the Realization phase.

5.1 FICS analysis

The FICS analysis, described in Section 3.8 serves to provide an overview of system usage from the system's point of view. The FICS analysis is split up into Functions and events, Interaction and Usability, Content and Structure, and Style and Aesthetics.

Functions and Events

There are five central functions/processes to this algorithm:

1. The process of filtering and preprocessing the data. This includes filtering any noise, artefacts and activities outside the range described in Section 4.8 (0.2 - 0.7 Hz).
2. Training the classification models on data recorded separately and compiled into moving data, stationary data, chest breathing and diaphragmatic breathing data.
3. Classify the data from the recording sessions in terms of "Diaphragmatic breathing", "Repetitive breathing" and "Movement".
4. Extracting features from the data. This includes breathing frequency per minute, the number of DBMs per day and the other such features listed in Section 4.8.
5. Compiling the aforementioned features and results of the classification in order to display them to the user using the Graphical User Interface designed by Florian Naumilkat.

Interaction and Usability

The system affords the user two types of interaction, active and passive. The user passively interacts with the system as they wear the Airleviate and go about their daily routine. The user actively interacts with the system when they view the feedback on the interface, and enter the threshold and buffer length values. Such a scenario can be seen visualized in the activity diagram in Section 5.3 below.

Content and Structure

The system will output features such as those listed in section 4.8 in the Preliminary Requirements.

Style and Aesthetics

Since there is no direct interaction between the user and the code behind the scenes of the graphical user interface, there is no aesthetic component to the program. Coding paradigms will be following such as snake_case for naming functions and camelCase for naming variables, as well as inline and overhead comments to explain sections of the code.

5.2 Cognitive Walkthrough

A visualization of how the data is processed in the program is found below in Figure 5.1, on the next page. To keep the image length short, the smallest amount of data to represent what happens in each process was used.

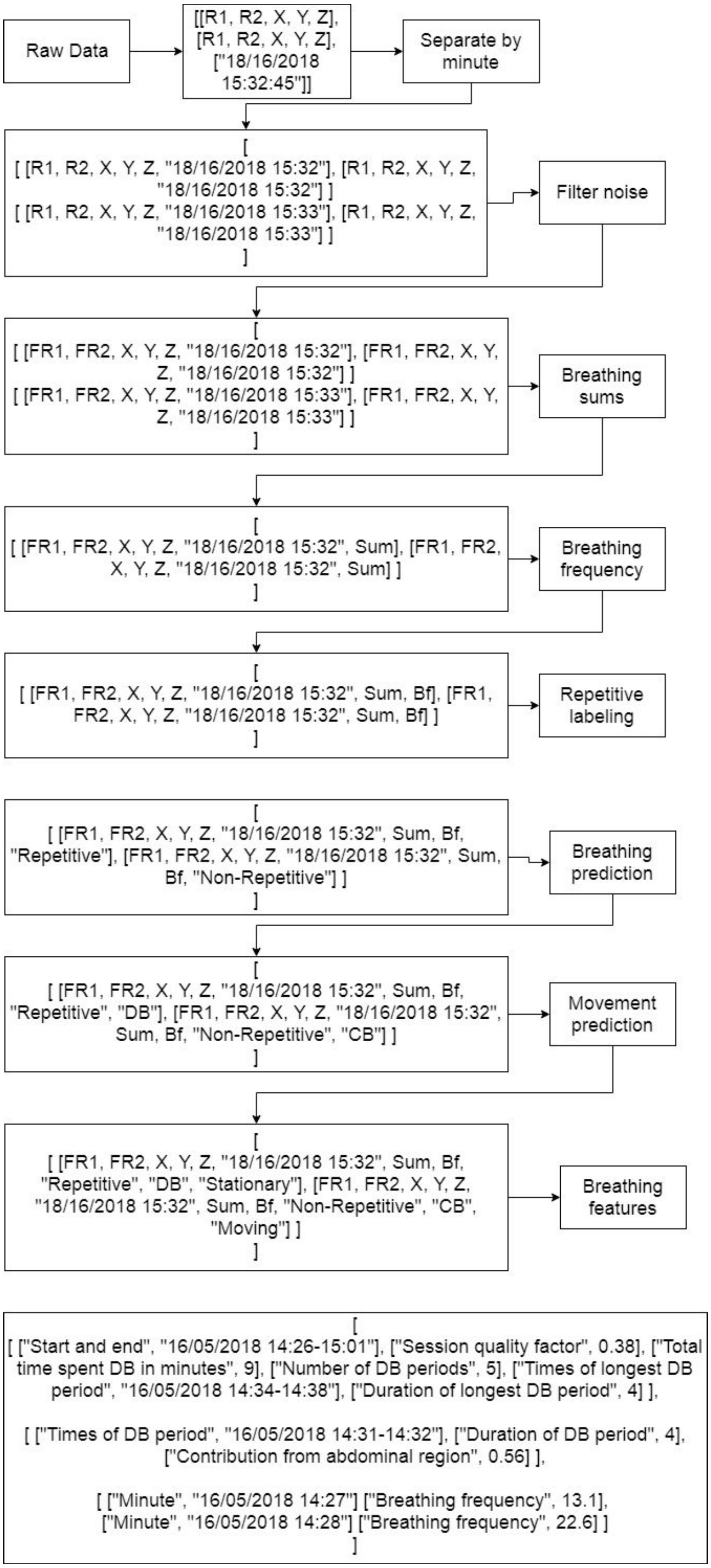

Raw Data
[[R1, R2, X, Y, Z], [R1, R2, X, Y, Z], ["18/16/2018 15:32:45"]]
Separate by minute
[[[R1, R2, X, Y, Z, "18/16/2018 15:32"], [R1, R2, X, Y, Z, "18/16/2018 15:32"]] [[R1, R2, X, Y, Z, "18/16/2018 15:33"], [R1, R2, X, Y, Z, "18/16/2018 15:33"]]]
Filter noise
[[[FR1, FR2, X, Y, Z, "18/16/2018 15:32"], [FR1, FR2, X, Y, Z, "18/16/2018 15:32"]] [[FR1, FR2, X, Y, Z, "18/16/2018 15:33"], [FR1, FR2, X, Y, Z, "18/16/2018 15:33"]]]
Breathing sums
[[[FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum], [FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum]]]
Breathing frequency
[[[FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum, Bf], [FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum, Bf]]]
Repetitive labeling
[[[FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum, Bf, "Repetitive"], [FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum, Bf, "Non-Repetitive"]]]
Breathing prediction
[[[FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum, Bf, "Repetitive", "DB"], [FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum, Bf, "Non-Repetitive", "CB"]]]
Movement prediction
[[[FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum, Bf, "Repetitive", "DB", "Stationary"], [FR1, FR2, X, Y, Z, "18/16/2018 15:32", Sum, Bf, "Non-Repetitive", "CB", "Moving"]]]
Breathing features
[[["Start and end", "16/05/2018 14:26-15:01"], ["Session quality factor", 0.38], ["Total time spent DB in minutes", 9], ["Number of DB periods", 5], ["Times of longest DB period", "16/05/2018 14:34-14:38"], ["Duration of longest DB period", 4]],
[["Times of DB period", "16/05/2018 14:31-14:32"], ["Duration of DB period", 4], ["Contribution from abdominal region", 0.56]],
[["Minute", "16/05/2018 14:27"] ["Breathing frequency", 13.1], ["Minute", "16/05/2018 14:28"] ["Breathing frequency", 22.6]]]

Figure 5.1: *Data transformation during the course of the program*

Before the program is run, the researcher needs to manually compile the training files for the various categories: “Diaphragmatic breathing”, “Chest breathing”, “Moving” and “Stationary”. When the program is run, it asks the user for the name of the raw data file, the name of the file to save the features to, the length of the buffer used to evaluate repetitive breathing and the tolerance level to classify diaphragmatic breathing with.

The raw data is loaded from the file and is checked to make sure there are no missing values from either band. The data is then separated by the minute, and then filtered.

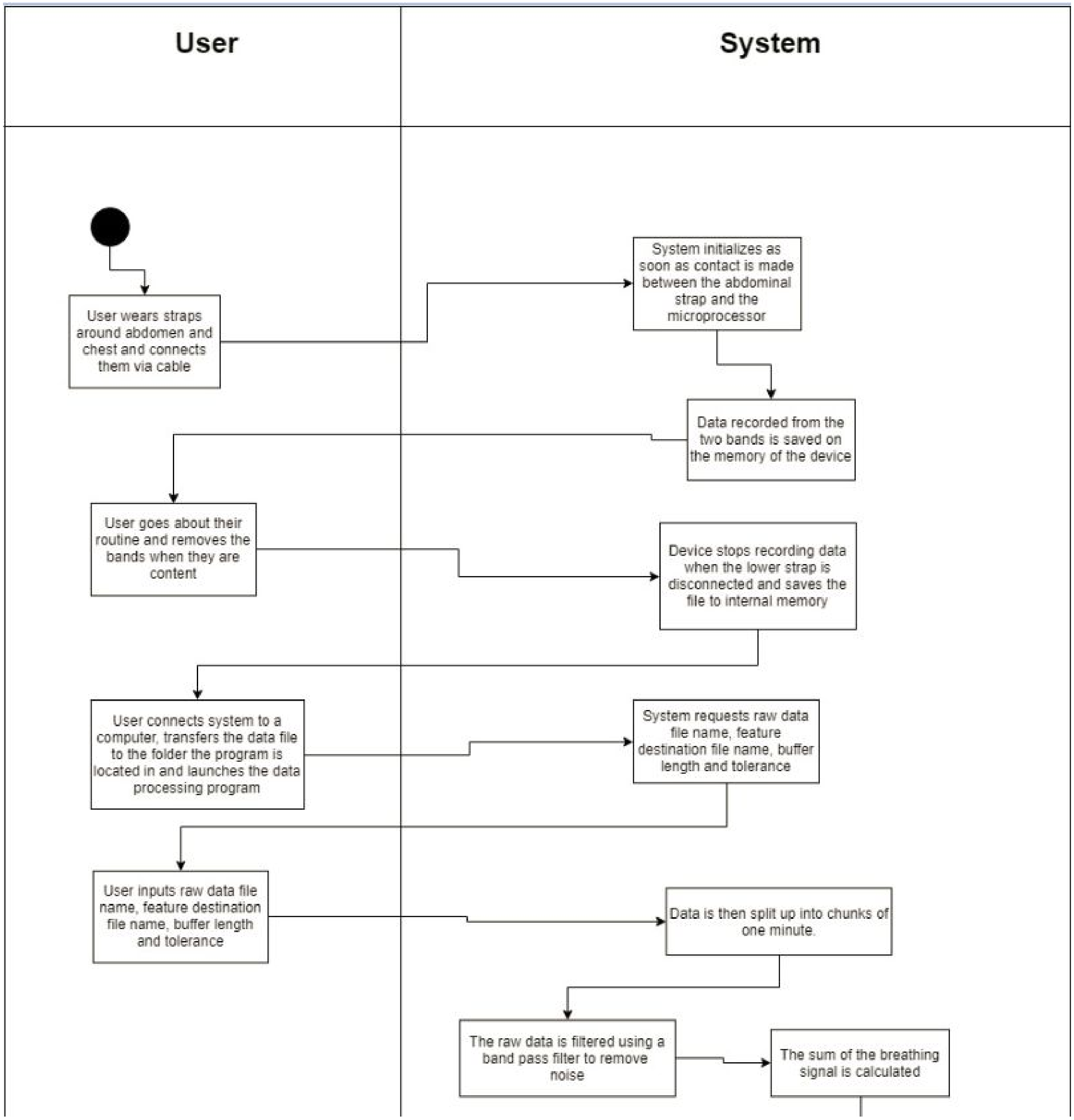
The sum of the breathing signal is calculated from the filtered data and appended to the data. This is then passed as an argument to the breathing frequency calculation component that appends the average breathing frequency of the entire minute to each sample in a minute.

This data is then passed to the repetitive breathing labeler, the diaphragmatic breathing classifier and then movement classifier.

Finally the breathing features are extracted from the data and written to the file specified by the user. A flow chart of how the data changes as it goes through the various components of the system is provided in Section 5.3. R1 and R2 correspond to the abdominal and chest band respectively; X, Y and Z represent the three axes of the accelerometer. FR1 and FR2 are the filtered R1 and R2 values.

5.3 Activity Diagram

This activity diagram in Figure 5.2, will show usage of the system in its current iteration. The final product will be much more user-friendly with calibration routines however currently, the process is slightly more complicated due to the early developmental phase of the system. The diagram shows the major actions fulfilled by the user and the system’s responses in order to understand the dynamic of the interaction. Therefore small actions like clicking the mouse button for example are left out.


User
System
User wears straps around abdomen and chest and connects them via cable
System initializes as soon as contact is made between the abdominal strap and the microprocessor
Data recorded from the two bands is saved on the memory of the device
User goes about their routine and removes the bands when they are content
Device stops recording data when the lower strap is disconnected and saves the file to internal memory
User connects system to a computer, transfers the data file to the folder the program is located in and launches the data processing program
System requests raw data file name, feature destination file name, buffer length and tolerance
User inputs raw data file name, feature destination file name, buffer length and tolerance
Data is then split up into chunks of one minute.
The raw data is filtered using a band pass filter to remove noise
The sum of the breathing signal is calculated

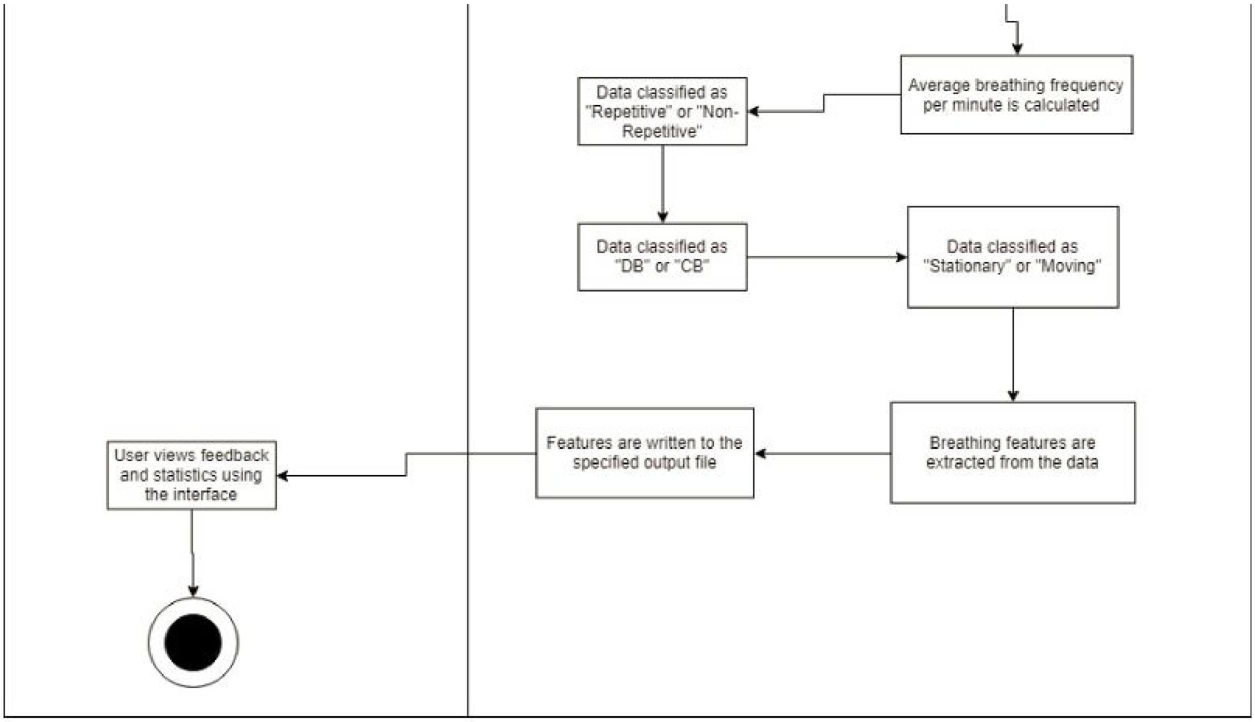

Data classified as "Repetitive" or "Non-Repetitive"
Average breathing frequency per minute is calculated
Data classified as "DB" or "CB"
Data classified as "Stationary" or "Moving"
User views feedback and statistics using the interface
Features are written to the specified output file
Breathing features are extracted from the data

Figure 5.2: *Activity Diagram*

5.4 System Architecture

The System Architecture provides an overview of the inner-workings and relations between various system components. This top down view of the system is accompanied by arrows connecting the various components, and can go a layer deeper to explore the workings of specific components, should they be comprised of more than one sub-component. As mentioned in Section 3.9, the System Architecture can be broken down into several levels, each of which is described below.

5.4.1 Level 0

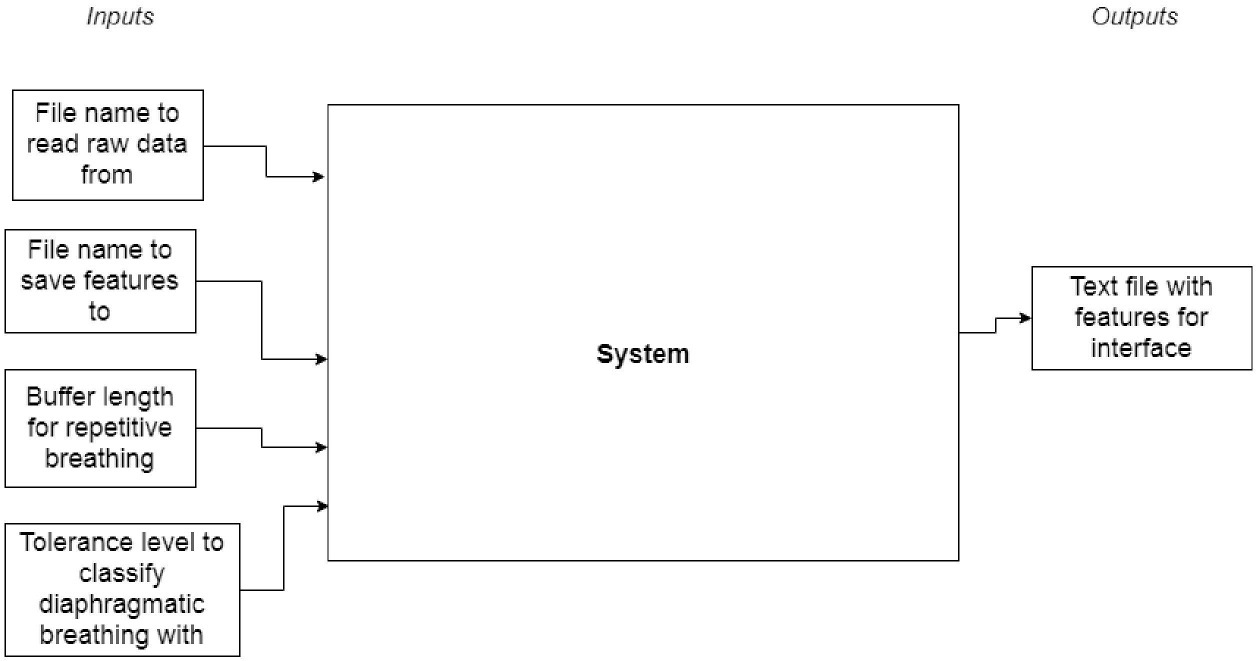

Inputs
Outputs
File name to read raw data from
File name to save features to
Buffer length for repetitive breathing
Tolerance level to classify diaphragmatic breathing with
System
Text file with features for interface

Figure 5.3: *System Architecture - Level 0*

This level consists of the inputs and outputs of the system, shown in Figure 5.3. The inputs to the system are the names of the file to read the raw data from, and the file to save the features to; the length of the buffer in minutes for the repetitive breathing moments and the tolerance with which to classify the diaphragmatic breathing. The name of the raw data file will be used to retrieve the data from the RIP bands and accelerometer. The accelerometer data will be used to help identify moments of rest, where the user’s position is relatively stationary. The RIP band data is used to measure contraction of the abdominal and thoracic (chest) compartments to measure breathing. The outputs of the system will be a list of features for the interface listed in Section 5.6. The raw data will be read from the filename specified and likewise, the features will be saved to the destination filename specified. The raw data file contains the RIP and accelerometer data from a particular breathing session.

5.4.2 Level 1

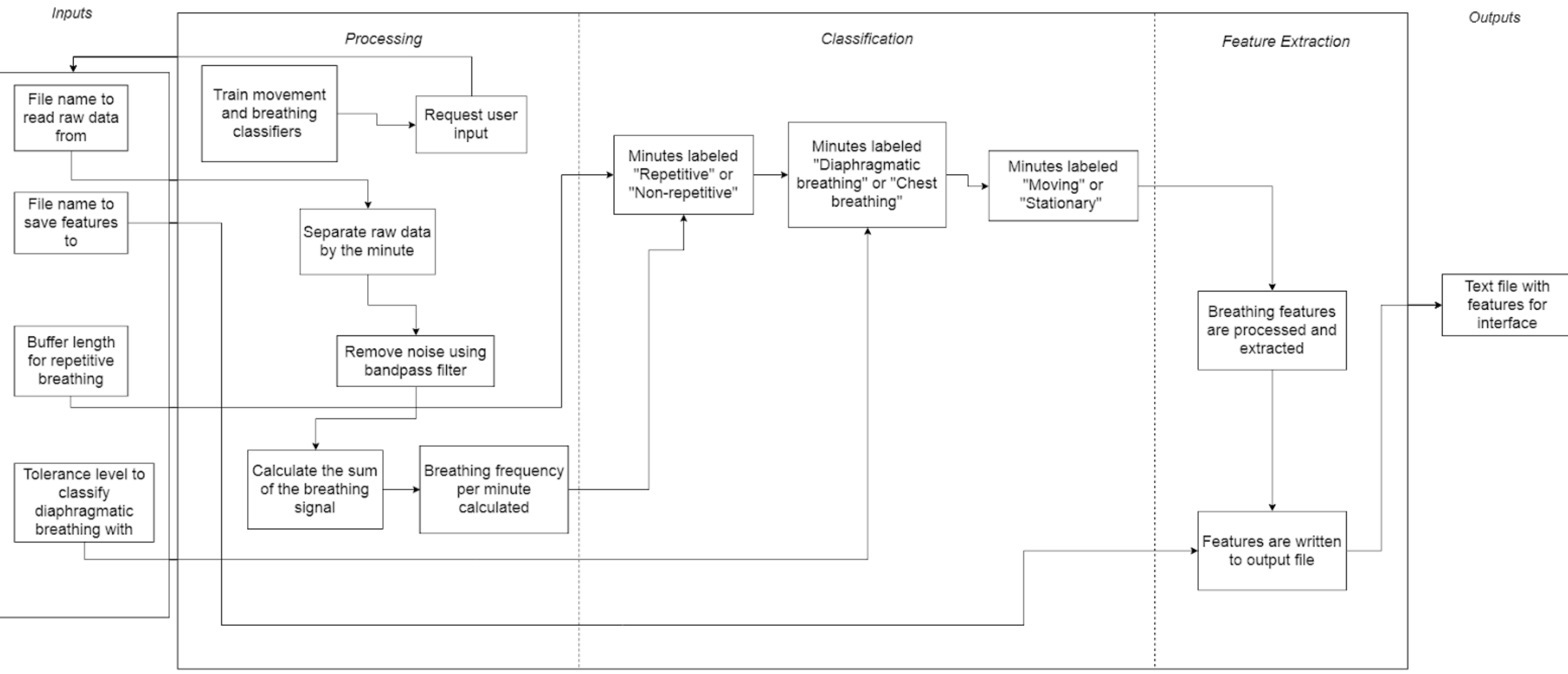

Inputs
Processing
Classification
Feature Extraction
Outputs
File name to read raw data from
File name to save features to
Buffer length for repetitive breathing
Tolerance level to classify diaphragmatic breathing with
Train movement and breathing classifiers
Request user input
Separate raw data by the minute
Remove noise using bandpass filter
Calculate the sum of the breathing signal
Breathing frequency per minute calculated
Minutes labeled "Repetitive" or "Non-repetitive"
Minutes labeled "Diaphragmatic breathing" or "Chest breathing"
Minutes labeled "Moving" or "Stationary"
Breathing features are processed and extracted
Features are written to output file
Text file with features for interface

Figure 5.4: *System Architecture - Level 1*

The first level, shown in Figure 5.4, consists of the overview of the system components. In this case, these components are descriptions of the steps that are taken to process the data, classify the data and extract the features. The diagram is therefore split up into the sections of Processing, Classification and Feature Extraction.

Before the raw data is processed, the classifiers are trained on data recorded while the participant was specifically performing diaphragmatic and chest breathing. This data is recorded during the session while the participant is instructed to maintain two specific postures: sitting and standing. The data is manually separated by the researcher, and is saved in separate files before launching the program.

Processing

For the data processing, first the data is separated by the minute using the timestamp provided in the raw data file. After this, noise and other artefacts is removed from the data by applying a band pass filter in order to get rid of medium-high frequency as well as baseline drift (very low frequency noise). The breathing signal sum for each sample is calculated and appended to it. The breathing frequency of a minute is averaged and appended to the end of each sample belonging to that minute.

Classification

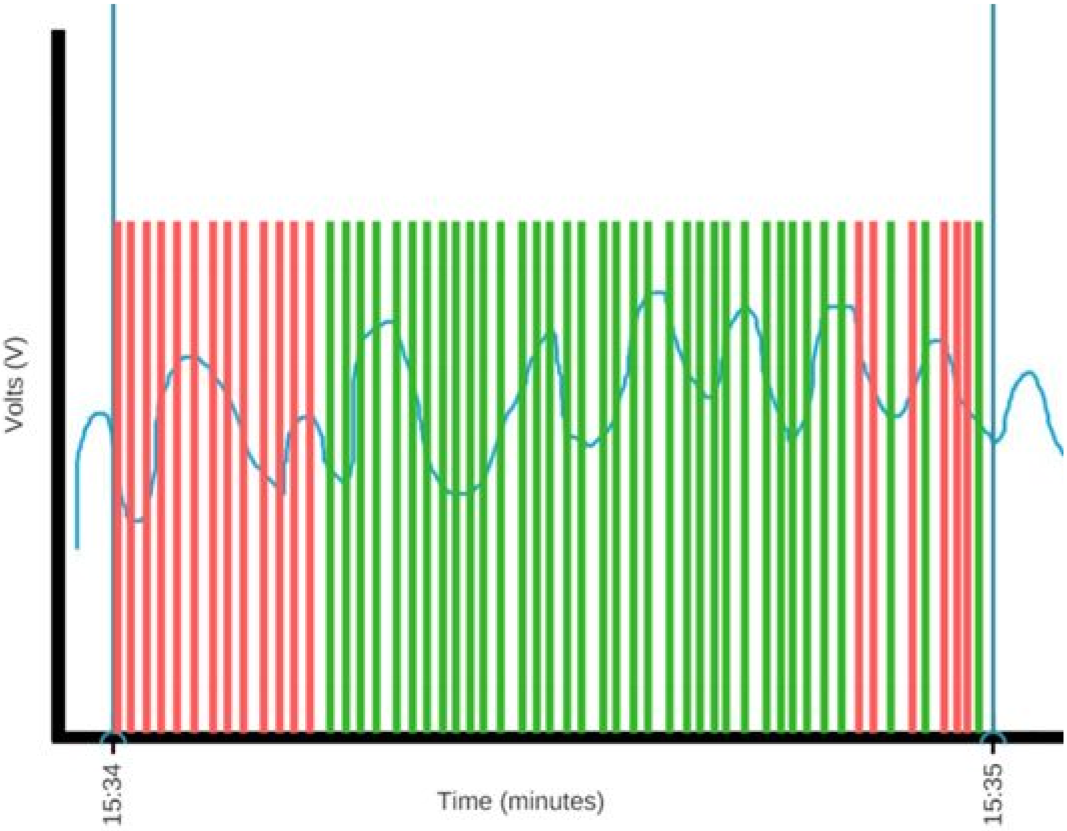


Figure 5.5: *Visualization of how the samples in a minute are classified.*

Repetitive breathing

The data is first analyzed for periods of “repetitive breathing”. This is done by checking whether the breathing frequency of the current minute falls within one standard deviation of the average breathing frequency of the minutes in the buffer. If it does fall within one standard deviation, then every sample from that minute is has the label “Repetitive” appended to the end of it. Otherwise every sample gets “Non-Repetitive” appended to the ended of it.

Diaphragmatic breathing

The breathing classifier will be fed a set of values relating to the RIP signal of each band. For each sample, the classifier will examine the values it is presented and predict either “Diaphragmatic” or “Chest”, a visualization of which is given above in Figure 5.5 where green represents diaphragmatic and red represents chest breathing. The sample then gets that label appended to the end of it. After this, the number of samples with the label “Diaphragmatic” is counted as well as the total number of samples in one minute. A ratio is calculated of diaphragmatic samples to total samples and is compared with the entered threshold value. If the ratio is higher, then every sample in that minute of data has the label “Diaphragmatic” appended to the end of it, and otherwise the “Chest” label.

Movement

Movement classification, as mentioned above, is used to identify “restful” moments. The X, Y and Z values for every sample is fed to the movement classifier as input. The classifier then predicts a label, “Moving” or “Stationary”, and this label is appended to the end of each sample. A visualization of this is given above in Figure 5.5 where green represents moving and red represents stationary movement. A similar process to diaphragmatic breathing occurs, where the number of samples labeled “Moving” are tallied and is divided by the total number of samples in a minute. If this value is larger than 0.5, the samples in that minute have the label “Moving” appended to their end, and otherwise “Stationary” appended to their end.

Extraction

During extraction, features mentioned in the Content and Structure section of the FICS analysis in Section 5.1 will be extracted from the data and saved to a file, which is entered by the user when the program starts.

5.4.3 Level 2

In this level, the individual functionality of components comprised of multiple subcomponents will be described. The “Breathing features are processed and extracted” component, shown in Figure 5.4, is complex, with multiple subcomponents and will be further explained below.

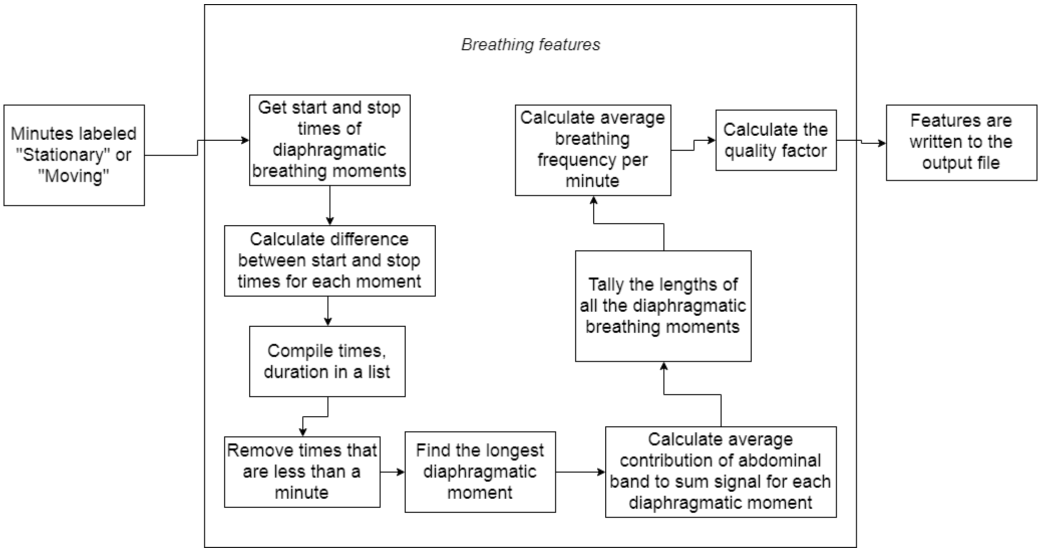

Breathing features
Minutes labeled "Stationary" or "Moving"
Get start and stop times of diaphragmatic breathing moments
Calculate difference between start and stop times for each moment
Compile times, duration in a list
Remove times that are less than a minute
Find the longest diaphragmatic moment
Calculate average contribution of abdominal band to sum signal for each diaphragmatic moment
Tally the lengths of all the diaphragmatic breathing moments
Calculate average breathing frequency per minute
Calculate the quality factor
Features are written to the output file

Figure 5.6: *System Architecture - **“**Breathing features are processed and extracted” component*

Breathing features are processed and extracted

The breathing features consists of an array with three subarrays. The first subarray lists the statistics of the recording session such as the start and end times of the recording, the quality factor of the session, the total time spent breathing diaphragmatically, the longest DBM and the times of this moment. The second subarray consists of the times of each diaphragmatic period, the length of each period and the average contribution of the abdominal band for the period. The final subarray is an array of the minutes and their corresponding average breathing frequencies.

“Get start and stop of diaphragmatic breathing moments” is the first subcomponent run which detects periods of diaphragmatic breathing by checking the label of each sample as well as a boolean to check whether currently in a period or not. The sample’s time stamp is added to the start-times list if the boolean evaluates to false and if the time isn’t already in the list. Similarly, the sample’s time stamp is added to the end-times if the sample’s breathing label is “Chest Breathing” and the boolean is set to false. The lists of the start and stop times are of equal length and one index corresponds to a start and stop time pair. “Calculate difference between start and stop times for each moment” accesses the start and stop times by their index and calculates the time difference between them in minutes to get the duration. The times and the duration for each moment are then compiled into a list in the “Compile times and durations in a list” process. “Remove times that are less than a minute” removes periods that are less than a minute as those are not considered.
“Find the longest diaphragmatic moment” goes through the list of periods and saves the longest to report as one of the statistics in the feature list. “Calculate average contribution of abdominal band to sum signal for each diaphragmatic moment” gets the start and stop times of each diaphragmatic moment and accesses the data passed in as an argument to calculate the average abdominal contribution to the sum signal during these times and appends it to the list of times and durations of diaphragmatic moments. “Tally the length of the diaphragmatic breathing moments” runs through the list of DBMs to add up the durations to report as a feature. Finally the quality factor is calculated by fetching the total recording time and dividing the duration of diaphragmatic breathing by it.

“Calculate average breathing frequency per minute” populates the third subarray by extracting all the individual minutes from the input data and calculating the average breathing frequency from each sample in a minute of data. This is then appended to the minute and added to the subarray.

5.5 Terminology and analysis decisions

This section will discuss decisions made during the development of the program to analyze, classify and process the data.

Diaphragmatic breathing

A minute of data is considered when classifying the breathing, as it gives ample time for at least 4-5 breathing cycles to occur during this window. Ma et al., (2017) found that adults could be trained to reduce their breathing rate to 4 breaths per minute with 8 weeks of training using live feedback during the training sessions, therefore considering the scope of this project, one minute will be sufficient to capture at least 7-8 breaths, if not more.

A user is considered to be employing diaphragmatic breathing when in a minute's worth of data, the majority of the breaths taken in that minute are classified by the SVM as diaphragmatic breathing. The total number of samples that are labeled diaphragmatic is tallied, as well as the total number of samples and the ratio is calculated. If this ratio is above the inputted tolerance level at the beginning of the program, the minute is declared diaphragmatic and every sample in that minute is labelled so. Therefore, after each pair of data points from the bands for every sample in a minute of the recording has been classified, if the number of samples labeled "Diaphragmatic breathing" exceeds the inputted threshold, that minute of breathing is defined as diaphragmatic and the samples within it are labeled as such. This process can be visualized by referring to Figure 5.5, where the green lines signify diaphragmatic breathing and the red lines signify chest breathing. If the ratio of diaphragmatic samples is higher than the tolerance, then all the samples are labeled diaphragmatic (marked green).

Accelerometer data

The decision to observe the accelerometer data in parallel to the breathing data was made as the focus shifted from moments of rest (discerned from accelerometer data) to moments of repetitive breathing. Now the accelerometer data is no longer used to filter data, but is classified and presented to the user to provide them with an overview of their movement activity and to put their breathing statistics into context.

While processing the accelerometer data, the average over the minute was taken to get a snapshot of the motion of the user for that minute. However, in averaging the accelerometer data over an entire minute, valuable movement information was lost. Any small movements and shifts in position were lost in the average and the data no longer accurately described the motion of the user. Even though movement data was specifically recorded, it was still classified as stationary due to the averages. Therefore, a similar approach to the diaphragmatic breathing classification method was adopted where every sample is classified as "Stationary" or " Moving", the number of stationary or moving moments are counted for the minute, but instead, the majority sets the movement classification for that minute.

So if the user’s movement was classified as stationary for the majority of the samples pertaining to a single minute, every sample in that minute is labeled “Stationary”. Figure 5.5 can be used to visualize this.

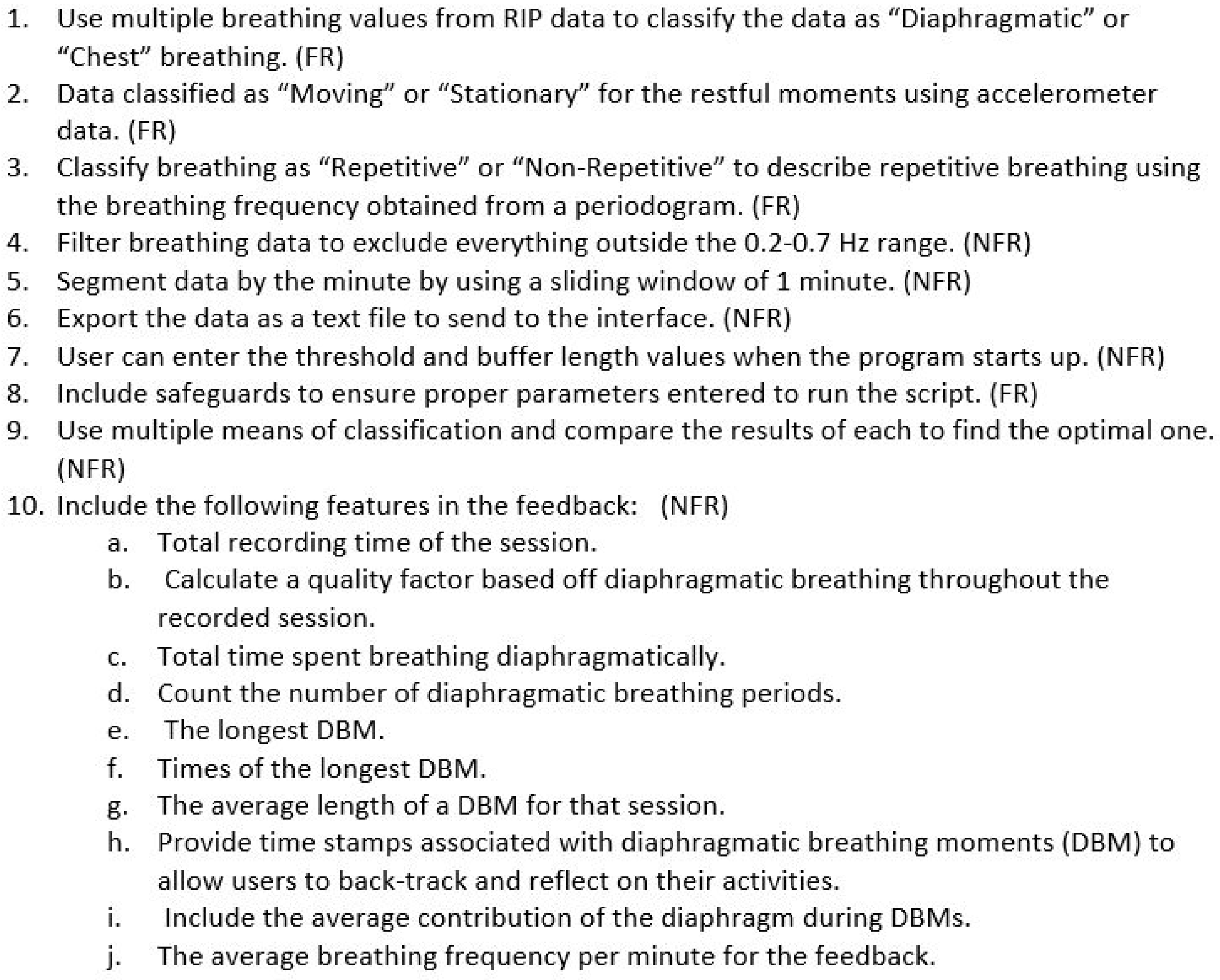
Repetitive breathing

The term “Repetitive breathing” was coined for when the users breathing does not vary too much over time. The breathing frequency of the user is used to measure this. Breathing was said to be repetitive when the average breathing frequency of the samples in a minute of data, falls within one standard deviation of the average breathing frequency of the minutes in the buffer. The length of the buffer in minutes is specified as an input to the program. Repetitive breathing is presented to the user in a similar fashion to the accelerometer data, for the purpose of reflection on activities during the day.

5.6 Requirements analysis: MoSCoW and Functional - Non-Functional

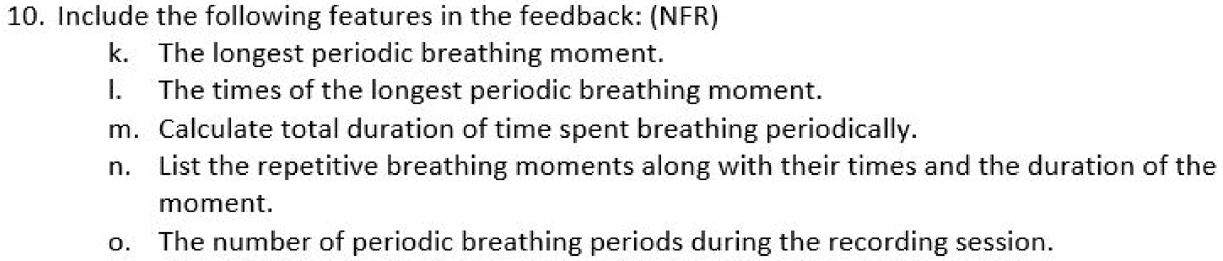
The MoSCoW method as described in Section 3.7 will categorize the Preliminary Requirements into Must, Should, Could and Won't have. Some of the requirements have changed in part due to the Expert Review or other revelations and shall be italicized. Further within this analysis, the labels (FR) and (NFR) will be placed behind the requirement signifying Functional and Non-Functional requirement respectively.

Must Have:



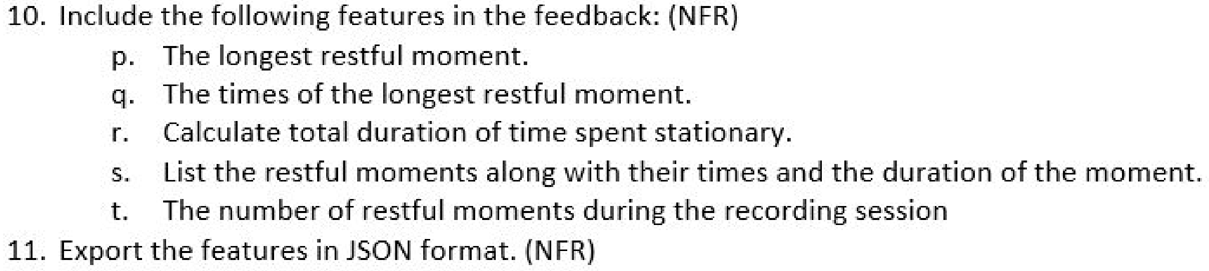
1. Use multiple breathing values from RIP data to classify the data as "Diaphragmatic" or "Chest" breathing. (FR)
2. Data classified as "Moving" or "Stationary" for the restful moments using accelerometer data. (FR)
3. Classify breathing as "Repetitive" or "Non-Repetitive" to describe repetitive breathing using the breathing frequency obtained from a periodogram. (FR)
4. Filter breathing data to exclude everything outside the 0.2-0.7 Hz range. (NFR)
5. Segment data by the minute by using a sliding window of 1 minute. (NFR)
6. Export the data as a text file to send to the interface. (NFR)
7. User can enter the threshold and buffer length values when the program starts up. (NFR)
8. Include safeguards to ensure proper parameters entered to run the script. (FR)
9. Use multiple means of classification and compare the results of each to find the optimal one. (NFR)
10. Include the following features in the feedback: (NFR)
    a. Total recording time of the session.
    b. Calculate a quality factor based off diaphragmatic breathing throughout the recorded session.
    c. Total time spent breathing diaphragmatically.
    d. Count the number of diaphragmatic breathing periods.
    e. The longest DBM.
    f. Times of the longest DBM.
    g. The average length of a DBM for that session.
    h. Provide time stamps associated with diaphragmatic breathing moments (DBM) to allow users to back-track and reflect on their activities.
    i. Include the average contribution of the diaphragm during DBMs.
    j. The average breathing frequency per minute for the feedback.

Should Have:



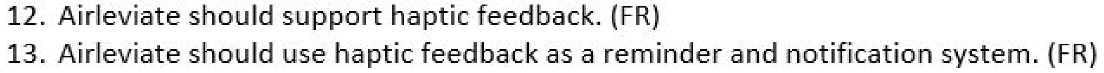
10. Include the following features in the feedback: (NFR)
    k. The longest periodic breathing moment.
    l. The times of the longest periodic breathing moment.
    m. Calculate total duration of time spent breathing periodically.
    n. List the repetitive breathing moments along with their times and the duration of the moment.
    o. The number of periodic breathing periods during the recording session.

Could Have:



10. Include the following features in the feedback: (NFR)
    p. The longest restful moment.
    q. The times of the longest restful moment.
    r. Calculate total duration of time spent stationary.
    s. List the restful moments along with their times and the duration of the moment.
    t. The number of restful moments during the recording session
11. Export the features in JSON format. (NFR)

Won't Have:



12. Airleviate should support haptic feedback. (FR)
13. Airleviate should use haptic feedback as a reminder and notification system. (FR)

6. Realization

This chapter describes the process of developing the program that cleans, classifies and analyzes the breathing data. The first section will discuss the process of development and how the author went about recording data. The second section describes a prediction accuracy test for the various classification models when the classifiers are trained on various features of the breathing data. The chapter ends with the a decomposition of the major functions used in the program. The functions written to clean, classify and analyze the data can be found in Appendix 11 F.

6.1 Development

While developing the program, the author constantly tested it on data recorded by himself in order to verify the functionality of the code. To train the classifiers, the author recorded 10 minutes of data while in the seated position and 10 minutes of data in the standing position. Of those 10 minute sessions, 5 minutes were spent breathing diaphragmatically and the other 5 were spent breathing primarily using the thorax (upper chest). The accuracy of various classifiers is investigated and compared in Appendix 11 D. with the combination of features that yielded that highest accuracy is covered in Section 6.2. Scikit-learn, a popular machine learning library for python, will be utilized for the classification algorithms used in this project (Scikit-learn.org, 2018). The author recorded a smaller ratio of training data compared to testing data in order to model a real world situation as closely as possible. If this product was to be released on the market, the users could not be expected to sit through hours of sessions where they are to record training data. The calibration routines would be short and easy. Therefore the models were trained with relatively little training data to the ratio of testing data (recording sessions).

The author recorded his own breathing spanning hours as well as shorter amounts of time, where he was more aware of the type of breathing he was using, to see if this was reflected in the results of the classification. During recordings where the author deliberately used more diaphragmatic breathing, there was a positive relationship between the amount of time spent breathing diaphragmatically and the number of DBMs, as well as the quality factor.

A sliding window was used to analyze the raw data, where each frame was one minute long. The classification of the data can be visualized by the graph in Image 6.1 below.

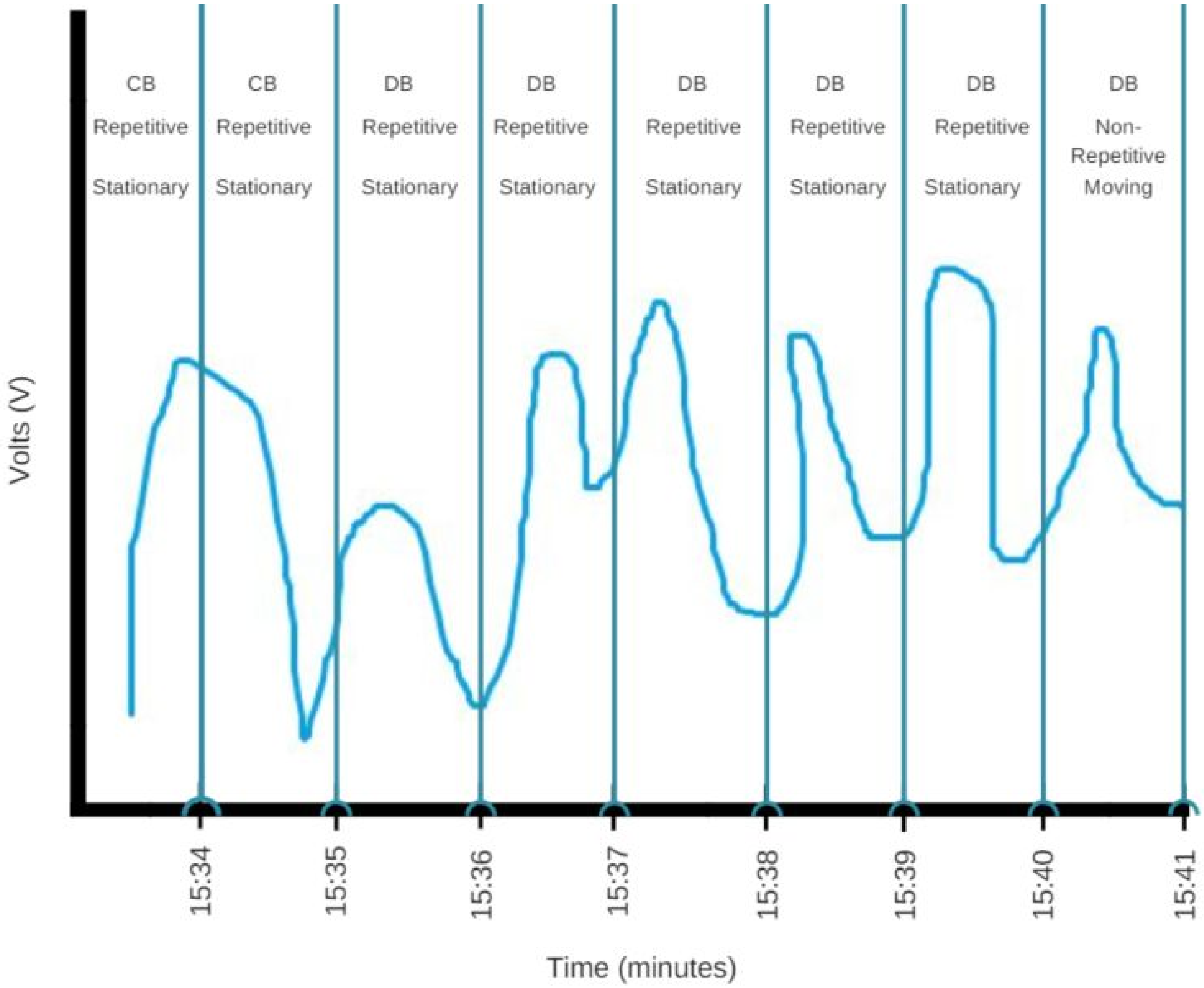


Image 6.1: *Visualization of how the data was analyzed and classified*

6.1.1 Main program functions

The implementation of the five main functions of the program as described in the Functions and Events in the FICS analysis in Section 5.1 are shown here.

The first function is to filter and preprocess the data. This includes filtering any noise, artefacts and activities outside 0.2 - 0.7 Hz. This is done by a function in the called *remove_noise(data)*, which takes the argument of data which represents the RIP values from a breathing recording. It uses filtfilt function from a python library called SciPy to filter frequencies outside the band mentioned above. A similar function, *remove_noise_training_data(trainingData)* is used to remove noise and artefacts from the training data recorded.

Classification models are trained on data recorded separately that were compiled into files containing moving data, stationary data, chest breathing and diaphragmatic breathing data. The functions *get_trained_breathing_classifier(trainingData, labels)* and *get_trained_movement_classifier(trainingData, labels)* are used to return classifiers for breathing and movement respectively that are trained on the training data recorded. The labels argument to both functions is a list of labels like “Diaphragmatic” or “Chest” to allow the classify to relate each data sample with a label to teach itself.

Classify the data from the recording sessions in terms of “Diaphragmatic breathing”, “Repetitive breathing” and “Movement”. The functions get_breathing_prediction(clf, data, threshold, unfilteredData) and get_movement_prediction(clf, data) add the classification labels to the ends of each sample in the data variable provided as input. Both functions take a classifier as an argument, which are the trained classifiers from the previous functions. In the case of the breathing prediction function, it also takes the threshold level inputted at the beginning of the program and the unfiltered data. Unfiltered data is also included in the classification of breathing due to results in Section 6.2.

The fourth function is to extract features from the data. This includes breathing frequency per minute, the number of DBMs per day and the other such features listed in Section 4.8. The function *db_features(data)* gets the data, where each sample has all of its labels appended (diaphragmatic, repetitive and moving). The function checks for the times a moment begins and ends by looking at the diaphragmatic label, calculates the length of the duration of a DBM and finds the longest duration. It also compiles the breathing frequencies per minute. A full explanation of this can be found in Section 5.4.3.

The fifth and final function is that of compiling the aforementioned features and results of the classification in order to display them to the user using the Graphical User Interface designed by Florian Naumilkat. This function is carried out by *write_features_to_file(features, name)*, which takes the list of features as an argument as well as the name of the file that the features should be written to. If the file already exists in the folder, the program overwrites the file, and if the file does not exist, it creates a file with the name specified and writes the features to it.

6.2 Classifier accuracy tests

In order to choose the right classifier for each type of data, RIP and accelerometer, tests were carried out comparing their prediction accuracies. A function from the scikit-learn python library called train_test_split() is used in order to split the labeled data into training and testing data, where the size of the testing data can be specified as a parameter for the function. Considering that only 20 minutes of training data in total will be recorded during the program tests, and that the ratio of the training data to the testing data is quite low due to an hour of testing data being recorded, the test split will be taken as 0.9 or 90% of the training data.

The classifiers that are compared are a Support Vector Machine, a Neural Network, and a Logistic regression classifier. The Logistic regression classifier is the only linear classifier that is considered; the other classifiers belong to the non-linear category.

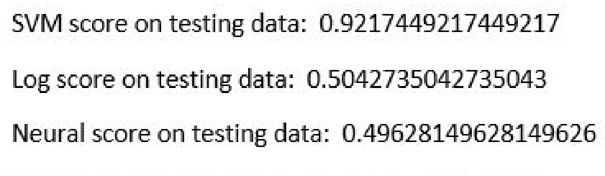
6.2.1 Breathing classifier test

For the breathing classifiers, different combinations of features were tested in order to find the combination that yields the highest prediction accuracy. The inclusion of extra features was a suggestion from Mrs. Garde, interviewed in Section 4.5 during the Expert Review. The features relevant to the RIP data are the raw values obtained from the two bands, their relative contribution to their sum signal and their filtered values. The breathing classifiers all have various parameters that they can be initialized with, for example, with a Neural Network, the number of layers in between the input and output layers can be specified. Due to the scope of this project, this test will use the classifiers on the basis of their default parameter values.

While testing the various classifiers, the function train_test_split from scikit-learn’s library shall be used to split the training data into training and testing data. In other words, this ratio represents the data from the original training set that will be used for testing. This ratio between training and testing data is specified as a parameter to the train_test_split function. Considering the ratio of training data to breathing session data is notably low, the data will be split with a ratio of 1:10 training to testing data.

Raw data, filtered data and relative contribution

All the features regarding the RIP data were compiled to train the classifiers. This means for each sample, the raw RIP values, the band pass filtered RIP values and the respective contributions to the RIP sum value. This combination yielded the highest accuracy for the SVM from all the tested combinations found in Appendix 11.D. Seeing as the SVM outperformed the other classifiers in classifying diaphragmatic breathing, it will be chosen as the classifier for diaphragmatic breathing and this combination of values will be used to train it.



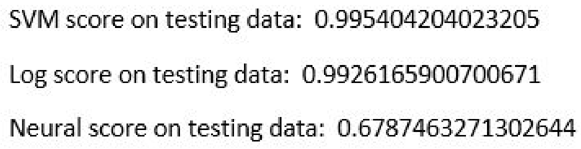
```
SVM score on testing data:  0.9217449217449217

Log score on testing data:  0.5042735042735043

Neural score on testing data:  0.49628149628149626
```

Image 6.2: *Breathing prediction scores of classifiers trained on raw data, filtered data and the relative contribution of each band to the sum*

Concluding from the tests above, the combination of the raw RIP values, band pass filtered RIP values and their respective contributions to the RIP sum value yields the highest prediction accuracy for all the classifiers. The highest prediction accuracy is achieved by the SVM classifier and will therefore be chosen to classify the breathing data.

6.2.2 Movement classifier test

The same classifiers tested for the breathing data will be used to classify motion. The classifiers are trained on the raw data coming from the accelerometer, which has been labeled according to the activity performed during the recordings by the researcher. The SVM and Logistic Regression analysis and both did exceptionally well in differentiating between “Stationary” and “Moving” data.



```
SVM score on testing data:  0.995404204023205

Log score on testing data:  0.9926165900700671

Neural score on testing data:  0.6787463271302644
```

Image 6.3: *Movement prediction scores of classifiers trained on raw motion data*

It was previously suggested by Mrs. Garde that a linear classifier should be sufficient to classify breathing data; however in light of these tests, and giving priority to the scenario where the amount of unlabeled testing data will outweigh the training data, the author decides to utilize an SVM for classification of both the breathing data and movement data.

It should be noted however, that the while recording the training data, the author made a conscious effort to distinguish between diaphragmatic and chest breathing. This slight exaggeration may have played a role in oversensitizing the classifier, so it may miss the more subtle breathing values that lie in between the two extremes of diaphragmatic and chest breathing.

6.3 Program decomposition

This section serves to provide the reader with an overview of the functions the constitute the System Architecture components shown in Figure 5.2, which is shown again on the next page.

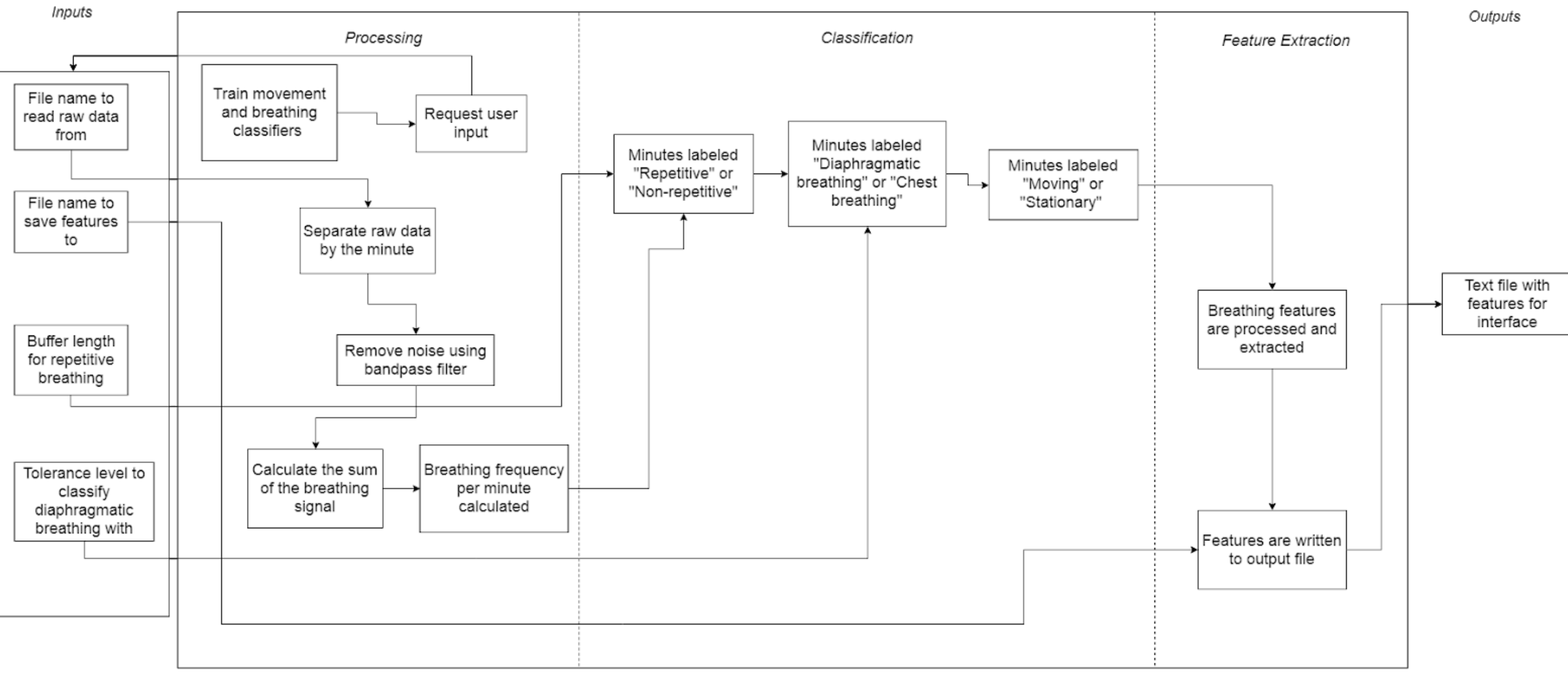

Inputs
Processing
Classification
Feature Extraction
Outputs
File name to read raw data from
File name to save features to
Buffer length for repetitive breathing
Tolerance level to classify diaphragmatic breathing with
Train movement and breathing classifiers
Request user input
Separate raw data by the minute
Remove noise using bandpass filter
Calculate the sum of the breathing signal
Breathing frequency per minute calculated
Minutes labeled "Repetitive" or "Non-repetitive"
Minutes labeled "Diaphragmatic breathing" or "Chest breathing"
Minutes labeled "Moving" or "Stationary"
Breathing features are processed and extracted
Features are written to output file
Text file with features for interface

Image 5.2: *System Architecture - Level 1*

The blocks that were already covered in Section 6.1.1 and in Section 5.4 will be skipped. Please refer to that section for an explanation.

Request User Input

This block contains the function *get_info(),* which queries the user for the raw data file name, the destination name for the features to be stored, the threshold value with which to classify diaphragmatic breathing and the buffer length in minutes for the repetitive breathing evaluation. This function has safeguards to prevent the user inputting data that may crash the program like entering a string value when the program expects an integer value in the case of the buffer length.

Separate raw data by the minute

This block contains the function *seperateData(data),* that takes the data set with timestamps distributed every 10 seconds between the samples, and returns a data set where the minutes have been found, and the data belonging to those minutes has been segmented into lists of samples pertaining to a minute of data.

Calculate the sum of the breathing signal

This block contains the function *get_breathing_sums(data)*, that takes the filtered data as input and for each sample in every minute of the data, calculates the sum of the sample and appends it to the end of said sample.

Breathing frequency per minute calculated

This block contains the function *breathing_frequency(data),* that takes the data with the sums appended to the end of each sample as an argument. The function looks at the data samples for every minute, and uses a periodogram from the SciPy python library to retrieve the powers of each of the frequencies found in the minute of data. The list of powers is then scanned for the highest power and the respective frequency for that power is found from list of frequencies outputted by the periodogram. This frequency is the appended to each sample in the minute of data and returned.

7. Evaluation

This section will discuss the steps taken to evaluate the program from a functional point of view and from the user's point of view. In section 7.1, various aspects of the program are tested and put through possible usage scenarios. Functionality like telling the time difference between two date-timestamps and filtering medium-high frequency noise from a signal are tested. Experts are then approached to comment on the methodology and approach to diaphragmatic breathing classification. Following that the program is tested in a "real" usage scenario to test if it picks up on planned DBMs using "experienced" diaphragmatic breathers. This is followed by a Functional Evaluation of the Revised requirements from Section 5.6. This section ends with a revised requirements list based on the results of the code tests, expert validations and "real world" test. As mentioned earlier, the term "Diaphragmatic Breathing moment" will be referred to as a "DBM".

This chapter will answer the following research sub-question:

"How can the validity of the classification be verified?"

By answering the last sub-research question, the author will be informed to answer to the main research question. This will be provided in the Conclusion (Section 8).

7.1 Code Testing

This section serves to partially answer this research sub-question by performing tests on specific components of the code to make sure they function as intended to, according to the requirements, and are able to handle various potential scenarios. Specific components of the code were tested in detail, the results of which are described below. Not every component was tested due to some of them being inherently tested when constructing other functions that relied on them to work properly, and thus it can be reasonably assumed that said components operate correctly. For example, a function that checks if a moment is diaphragmatic or not by accessing a sample's breathing label, is inherently tested while testing a function that extracts DBMs.

- The frequency filter was tested to make sure the correct frequencies, medium-high, were being filtered.
- The separation of data according to its time-stamp was tested to ensure all the data belonging to a minute was segmented as such.
- The time difference component was calculated to ensure that the amount of time in minutes between two timestamps was calculated correctly.
- The contribution of the diaphragmatic band to the sum signal was tested to ensure that this was being calculated properly and that the function could properly handle negative values.

- The function used to calculate the main frequency in the breathing signal was tested to ensure it extracted the correct frequency.
- The input tolerance was tested to find the realistic upper and lower bounds of the value. The input safeguards function was tested to ensure that the program did not crash when it received an input it was not expecting.
- The detection of DBMS was tested to ensure that the system reports DBMs properly.

7.1.1 Frequency Filter Test

One of the most crucial aspects of the code is the filtering of the data. Due to movement, shifting or talking, noise is introduced into the raw data signal. This noise needs to be removed in order to prepare the data for classification and analysis. During the realization phase, the Butterworth filter was used in "bandpass" mode to generate parameters for the filtfilt function. Breathing frequencies range from 0.2 to 0.7 Hz and so frequencies outside this range need to be removed. In order to calibrate the filter, the Nyquist frequency was calculated by dividing the sampling frequency (8 Hz) by two. The upper and lower bounds of the band (0.7 and 0.2) were then divided by the Nyquist frequency. These were then fed as parameters to the filter function.

The filtering function being used from the SciPy python library is the filtfilt function. Filtfilt applies a linear filter twice, once forwards and once backwards on the data. Observing Image 7.1, when the raw breathing data is filtered using the filtfilt algorithm, the data is normalized as well as missing erratic peaks in the area of the tag "1". The numbers in the Image 7.1 reflect the peaks; it can be seen from the graph of the filtfilt data (Image 7.1 - right), that the peaks correspond in position and amplitude to the raw data; the slight change in amplitude is due to the normalization of the values which is done by the filtfilt function.

Considering that filtfilt does a good job at isolating the desired frequency from the breathing signal while maintaining relative amplitude proportionality, the filtfilt function will be used to filter the breathing data. In addition it also normalizes the data, something recommended by Mrs. Garde in Section 4.5.

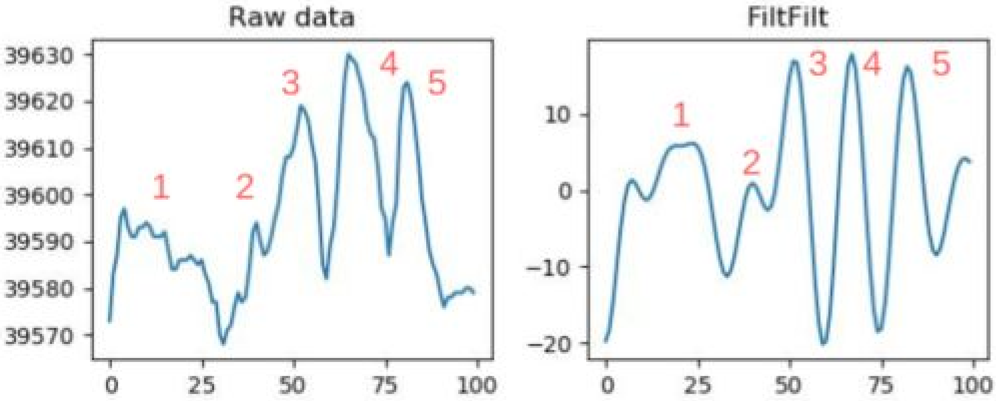
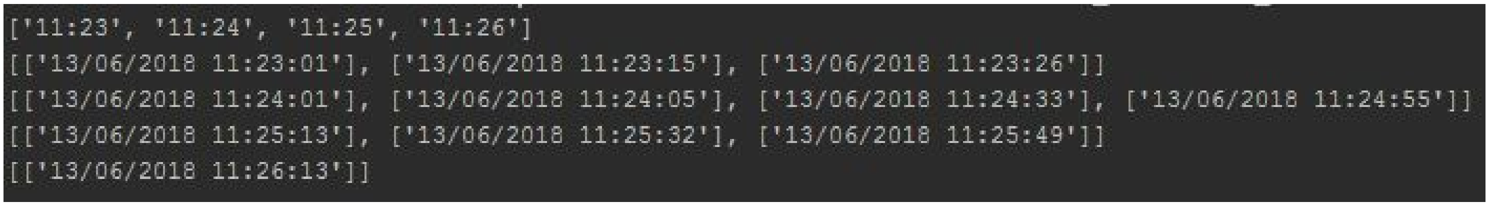


Image 7.1: *Comparison of the raw data (right) with filtered data using the filtfilt function (left)*

7.1.2 Time Separation Test

In order to test if this functionality was working, dummy data was created where date-timestamps were generated akin to those appearing in the raw data from the device. The code being tested was altered slightly for the test; the code in the program was dealing with data that had a different structure (sum signal, breathing frequency and labels for breathing and motion appended to it). The data the function was being tested on involved just the date-time stamp for simplicity. The code extracted a list of times from the data, and then compared the times of each sample to its collected list. The results can be seen in Image 7.2 above. The top line represents a list of the minutes the samples in the data belonged to. Underneath that, every line represents the data belonging to one minute. The data from each minute is compiled in to a list as shown in Image 7.2. In the second line, the three samples all belong to the date-time stamp “13/06/2018 11:23”. This shows that the code successfully segments the data by minute.

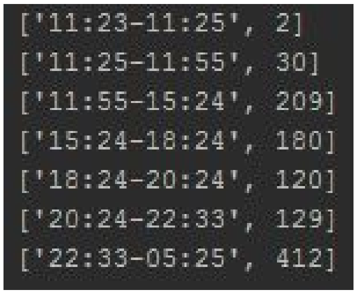


```
['11:23', '11:24', '11:25', '11:26']
[['13/06/2018 11:23:01'], ['13/06/2018 11:23:15'], ['13/06/2018 11:23:26']]
[['13/06/2018 11:24:01'], ['13/06/2018 11:24:05'], ['13/06/2018 11:24:33'], ['13/06/2018 11:24:55']]
[['13/06/2018 11:25:13'], ['13/06/2018 11:25:32'], ['13/06/2018 11:25:49']]
[['13/06/2018 11:26:13']]
```

Image 7.2: *Test results from grouping the data by minute.*

7.1.3 Time Difference Test

Calculating the amount of time that has passed between two times is a crucial functionality of the code. It is used to calculate the total recording time, the lengths of diaphragmatic breathing moments and the quality factor. The code was tested by giving arguments where the time span was contained in an hour, when the hour rolled over, when the times were contained in a day and when the day rolled over. The results are shown above in Image 7.3. The first column signifies the time period, and the second is the number of minutes between the two times. All cases were successfully handled by the code.

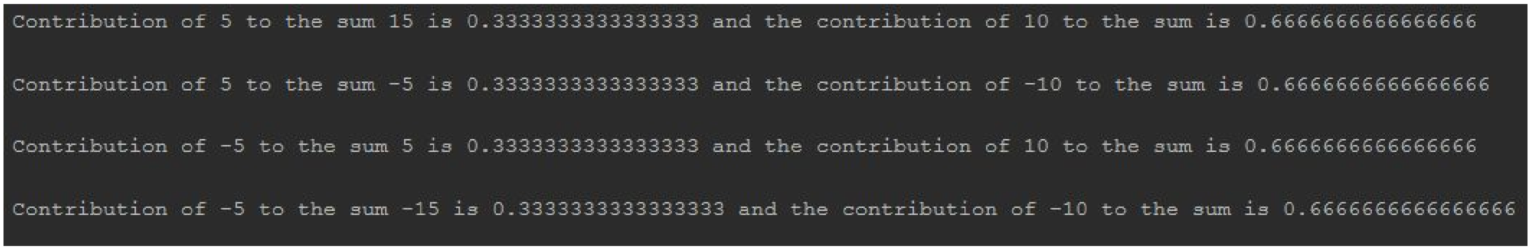


```
['11:23-11:25', 2]
['11:25-11:55', 30]
['11:55-15:24', 209]
['15:24-18:24', 180]
['18:24-20:24', 120]
['20:24-22:33', 129]
['22:33-05:25', 412]
```

Image 7.3: *Time periods and durations between each period*

7.1.4 Contribution Calculation Test

The contribution of the diaphragmatic band to the sum is fundamental as this is the part of the input that the breathing classifier uses to make its prediction. During the noise filtering, the data is normalized around 0, meaning some of the values become negative in value. Normally, calculating the contribution of a value to its sum is easy. However in this case, the inclusion of negative numbers gives rise to the event that the sum is less than one of the contributions is possible. The test will be run with three scenarios: the first where both numbers are positive, the second where one is negative and the last when both are negative (Image 7.4). The numbers used were 5 and 10 for the first scenario, 5 and -10 for the second scenario and -5 and -10 for the third scenario. The results are logical as the absolute values of the two parts were used in the calculation. This was to ensure that there were no errors when negative numbers are encountered. Assert statements were used in the code to ensure that if the absolute value of x was more than that of y, then its contribution should be higher than y’s and vice versa.



```
Contribution of 5 to the sum 15 is 0.3333333333333333 and the contribution of 10 to the sum is 0.6666666666666666

Contribution of 5 to the sum -5 is 0.3333333333333333 and the contribution of -10 to the sum is 0.6666666666666666

Contribution of -5 to the sum 5 is 0.3333333333333333 and the contribution of 10 to the sum is 0.6666666666666666

Contribution of -5 to the sum -15 is 0.3333333333333333 and the contribution of -10 to the sum is 0.6666666666666666
```

Image 7.4: *Results of the contribution*

7.1.5 Periodogram Test

A periodogram is a tool used in graphical data analysis of signals in the frequency domain. In particular it deals with signals that are equi-spaced, meaning the space between each point is constant. This applies to signals sampled at a certain constant frequency which is what the Airleviate does (samples at 8 Hz). A periodogram was used to detect the breathing frequency of a minute of data. The two signals from the bands were summed and used as input for the periodogram function. The function gave two lists as output; the first was a list of frequencies that were measured and the second list was the power of each frequency measured. The frequency with the highest power was assumed to be breathing frequency as noise outside the frequency band of interest had been removed using the Bandpass filter previously mentioned in Section 7.1.1. In order to test if the periodogram finds the dominant frequency in the breathing data the results of the periodogram were plotted and shown in Image 7.5. The image shows a spike in power at around 0.25 - 0.3 Hz. This is consistent with the breathing rate at rest while sitting, which is how the data for this particular test was recorded. A frequency of 0.3 Hz is equal to 18 breaths per minute, which is a normal rate for an individual at rest.

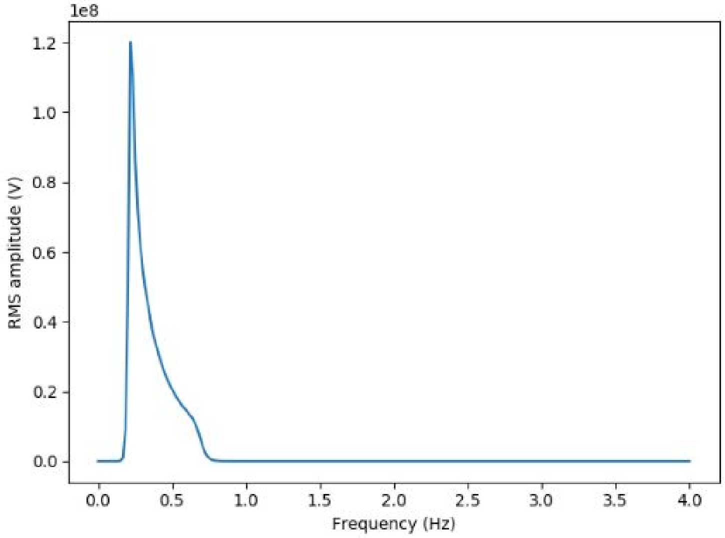


Image 7.5: *Plot of the powers of each frequency in the breathing signal in a minute of data.*

7.1.6 Input Tolerance Test

The tolerance level is requested as input when the program is launched. This decimal value is compared with the total diaphragmatic breathing ratio in a minute and if the diaphragmatic breathing ratio exceeds it, every sample in the minute is labeled diaphragmatic. The tolerance level needs to be at least 0.5, meaning that the diaphragmatic breathing ratio needs to be more than half of the minute's samples for it to be classified as diaphragmatic. If more than half of the samples in a minute are classified as "Chest breathing", then the wearer cannot have been said to have been using diaphragmatic breathing during that minute. Therefore, only if the number of samples in a minute labelled "Diaphragmatic" is greater than those labeled "Chest", i.e. greater than half, is the minute considered "Diaphragmatic".

This test will define a suitable range for the tolerance values; if the value is set too high, then the classification becomes very strict and the number of diaphragmatic breathing periods, the time spent breathing diaphragmatically and the quality factor all reduce noticeably. The quality factor is calculated by dividing the total duration of time spent breathing diaphragmatically by the total duration of time recorded, is used to evaluate the diaphragmatic breathing performance during a recording session.

For the purpose of testing, values ranging from 0.5 to 0.9 were used as input for the tolerance. Table 7.2 below shows the values of the features mentioned above relating to the inputted tolerance level. A breathing session was recorded where for the first half of the session, the author performed diaphragmatic breathing and for the latter part, performed chest breathing.

Tolerance level	Number of diaphragmatic periods	Total time spent breathing diaphragmatically	Quality factor
0.50	5	18	0.51
0.55	6	15	0.43
0.60	6	14	0.40
0.65	6	12	0.34
0.70	5	11	0.31
0.75	5	9	0.26
0.80	3	4	0.11
0.85	0	0	0.00

Table 7.2: *Results of testing various tolerance levels on the diaphragmatic breathing features.*

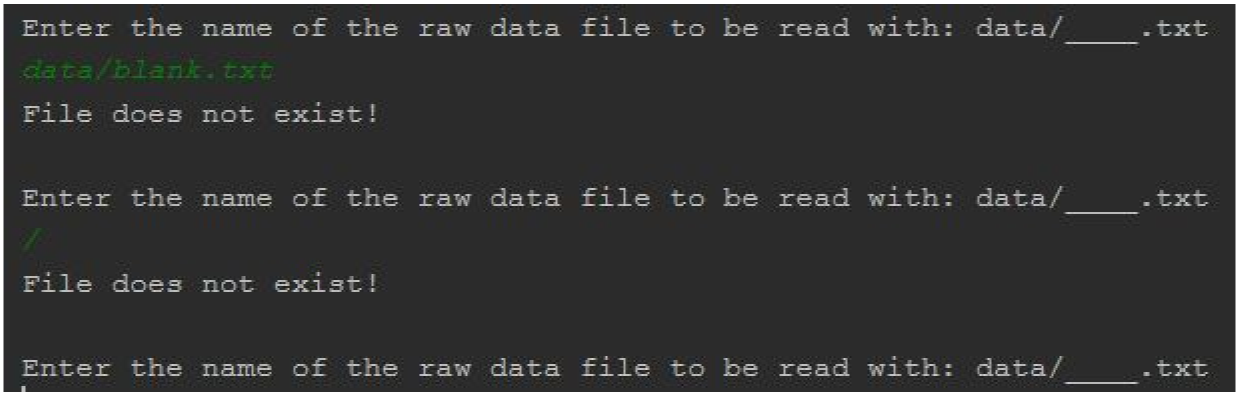
From Table 7.2 it can be seen that as the value rises above 0.80, the tolerance level is too high and therefore too strict for the classification of breathing. The SVM prediction accuracy, explored in Section 6.2.1, was shown to be 92%, meaning out of 100 samples, 8 will be wrongly classified. Technically the upper bound for the tolerance value could be considered 0.92, however as found from the test, values above 0.8 are far too strict. For this purpose, the range will be 0.50 to 0.8, although 0.8 is also a very strict threshold considering a quality factor of 0.11 was a result.

7.1.7 Input Safeguard Test

This function was put into place to avoid the program crashing in the scenario where a user inputs a value that the program is not expecting; for example a string value when the program is expecting a decimal or integer value. The program asks the user for four inputs: the name of the file to read the raw data from, the name of the file to output the breathing features to, the number of minutes to be used in the buffer for labeling repetitive moments and the tolerance level with which to label a minute with diaphragmatic or chest breathing. In the first and second instances, the program expects a string (a sequence of letters and numbers) as an argument. In the third and fourth instances, the program expects an integer value between 1 and 7 a and a float value between 0.50 and 0.99 respectively.

Raw data file name

The program ensures that the entered file name does not being with a “/” as this command can be used to navigate to the folder above where the code is located and is a security concern. The program also ensures that the entered name refers to a file that is in the data directory as shown in Image 7.6.

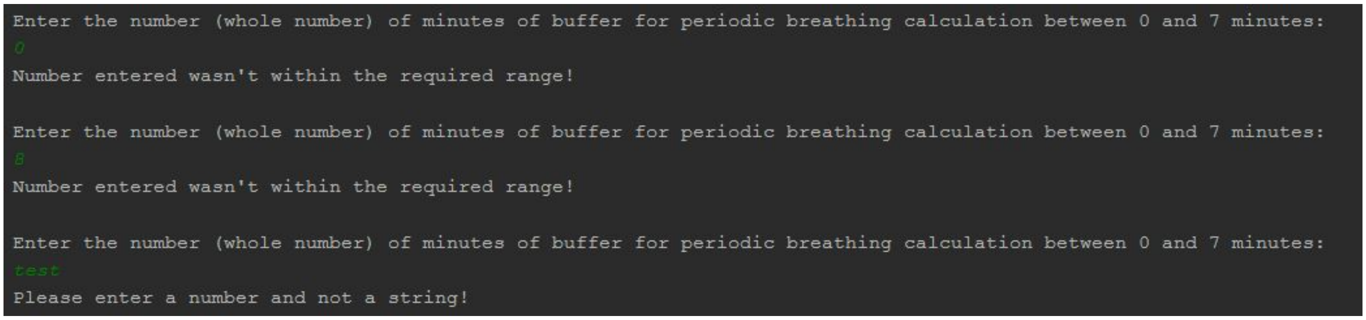


```
Enter the name of the raw data file to be read with: data/____.txt
data/blank.txt
File does not exist!

Enter the name of the raw data file to be read with: data/____.txt
/
File does not exist!

Enter the name of the raw data file to be read with: data/____.txt
```

Image 7.6: *Testing invalid file name*

Buffer input

The buffer, as explained earlier, is used to evaluate repetitive breathing. The buffer contains data of whole minutes, and therefore needs to be larger than zero. The upper-limit was set as seven minutes, the larger the limit, the more likely it is that the current minute's variance is to below the buffer's variance. The buffer input test is shown in Image 7.7.

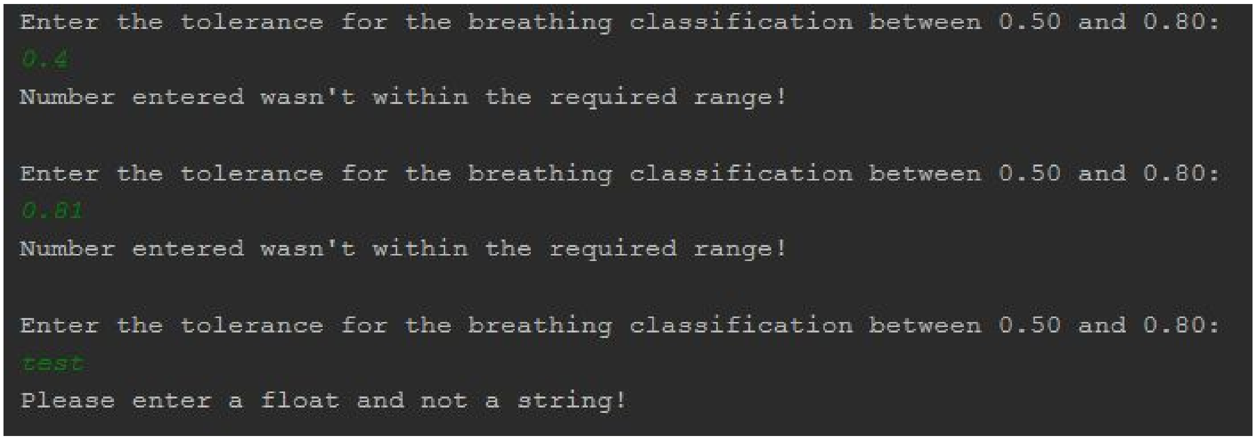


```
Enter the number (whole number) of minutes of buffer for periodic breathing calculation between 0 and 7 minutes:
0
Number entered wasn't within the required range!

Enter the number (whole number) of minutes of buffer for periodic breathing calculation between 0 and 7 minutes:
8
Number entered wasn't within the required range!

Enter the number (whole number) of minutes of buffer for periodic breathing calculation between 0 and 7 minutes:
test
Please enter a number and not a string!
```

Image 7.7: *Testing invalid buffer inputs*

Tolerance input

As discussed in section 7.2.6, the lower bound of the tolerance level is 0.5 and the upper bound is 0.8. If the tolerance is any higher, the feature extraction isn't descriptive of the breathing session. Lower than 0.5 and the classification becomes far too lenient. The check also ensures the input is in format of a float and not another type. The test is shown below in Image 7.8.



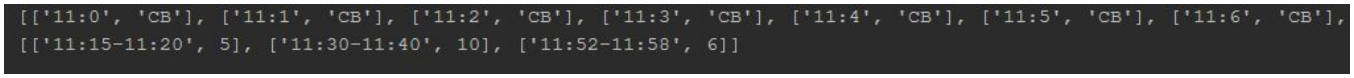
```
Enter the tolerance for the breathing classification between 0.50 and 0.80:
0.4
Number entered wasn't within the required range!

Enter the tolerance for the breathing classification between 0.50 and 0.80:
0.81
Number entered wasn't within the required range!

Enter the tolerance for the breathing classification between 0.50 and 0.80:
test
Please enter a float and not a string!
```

Image 7.8: *Testing invalid tolerance inputs*

7.1.8 DBM Detection Test

The DBM detection functionality was tested by creating dummy data with timestamps and labels of either “DB” or “CB” signifying diaphragmatic and chest breathing respectively. The author decided to have DBMs of various lengths tested. The times were chosen to be 11:15, 11:30 and 11:52, where a 5, 10 and 6 minute DBM should be detected. The results in Image 7.9 show that all DBMs were successfully detected and reported. The lower list in Image 7.9 represents the DBMs and their durations.

The outcomes of the code tests show that the functions are working as they were intended to and are capable of handling various possible scenarios of input including error handling.



```
[['11:0', 'CB'], ['11:1', 'CB'], ['11:2', 'CB'], ['11:3', 'CB'], ['11:4', 'CB'], ['11:5', 'CB'], ['11:6', 'CB'],
[['11:15-11:20', 5], ['11:30-11:40', 10], ['11:52-11:58', 6]]
```

Image 7.9: *Testing detection of DBMs*

7.2 Expert Validation

Two experts were approached to verify and validate the methodology and steps taken while developing this program. Dr. Kamilaris from the Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), who teaches a course on data visualization to second year Creative Technology students, was approached for the purpose of discussing the data processing and cleaning. Mrs. Garde from the Biomedical Signals and Systems group was approached to discuss the approach of classifying the breathing data.

7.2.1 Data usage and processing

The author of this report sought out an expert in the field of data visualization and processing to get an opinion on the way that the breathing and accelerometer data was processed in the program. The author met with Dr. Kamilaris and walked him through the data processing with the help of the Cognitive Walkthrough from Section 5.6.

Dr. Kamilaris approved of the methodology behind cleaning the signal (filtering) and dividing it per minute. With regards to the features extracted for data, he was in fact against presenting and overwhelming the users with graphs and numbers and instead liked the idea of a single measure of performance, which is the quality factor in this case. He pointed out the current trend in data visualization in the industry was very minimalistic, and meant to give the users a feel for the data.

For future work, Dr. Kamilaris recommended expanding the list of parameters that the classifiers are trained on to include breathing frequency, movement and repetitive breathing labels. He said it was wrong to assume that even if it seems logical that the breathing frequency should not affect the type of breathing one uses, there is no relation between the two. He did however mention that 92% classification accuracy (for breathing) was quite a high value in and of itself. Finally, Dr. Kamilaris brought up the fact that in testing the classifier, the author recorded data that he believed to belong to a certain label. However, there could be errors in that data due to small shifts, slippage or irregularity that can corrupt the classifiers categorization of the phenomenon.

In other words, if a number of chest breaths were accidentally performed while recording diaphragmatic breathing to train the classifier, the classifier takes this data at face value and assumes it to belong to the label provided. Therefore a classifier trained on this data may classify a chest breath as a diaphragmatic breath because of the corrupted training data. The difficult part he said is to find the proper data to train the classifiers on.

Dr. Kamilaris was also particularly interested in the “transfer knowledge” aspect of the program i.e. how the model developed in the program can be applied to the classification of other users’ breathing data. He said this would be an interesting aspect to investigate in further research. Another area he mentioned to look into was confusion matrices, to see where the classifier went wrong and why it went wrong. This is helpful in explaining strange classification errors or inaccurate breathing features. Finally, Dr. Kamilaris explained that it would be useful to export the weights that the classifiers uses for the various parameters in order to find out which are the most relevant features. These are both ideas for future work.

7.2.2 Breathing classification

Mrs. Garde, the expert who helped review the requirements in Section 5.2 was approached for her opinion on the implementation. After explaining Figure 5.5 and the Cognitive Walkthrough in Section 5.2,
Mrs. Garde approved the methods used for classification of the breathing data as well as the way the classification was approached (in terms of considering 1 minute at a time).

Mrs. Garde also mentioned how during the classifier prediction accuracy tests, K fold cross validation could be used instead of train_test_split and explained the difference between the two. The K in K fold cross validation, signifies the number of times the training dataset is split. The models are trained with one portion of the data and tested on the rest. They are then trained on the other portions of the data and tested on the rest. This repeated procedure gives a more realistic view of how the classifier will perform when handling “real world” situations, than compared to the simple train test split which only splits the data once.

Concerning the movement classification, Mrs Garde mentioned that the sum of the accelerometer axes can be used as an additional feature for training the movement classifier. The axes should be processed in the following way:

Sum = square_root(x^2 + y^2 + z^2)

She also mentioned that if any filtering is done on the accelerometer data, it should be done on the individual axes before the summation.

Mrs. Garde was pleased with the way the classification of breathing data was approached and approved the methodology. The suggestions she made regarding K fold cross validation and the additional accelerometer feature are points for future work.

7.3 Program Verification Test

“How can the validity of the program be verified?”

This section helps answer the first question by performing measurement tests with experienced diaphragmatic breathers. The participants are asked to breathe using their chest for 2 and a half minutes and using Diaphragmatic breathing for 2 and half minutes. Both these styles of breathing done while sitting first, and then while standing. Following this, 10 minutes of movement data is recorded where the participant is asked to walk around the room. The participants are then be asked to wear the Airleviate for one hour and to prepare a written list of times beforehand, at which point they will pay special attention to their diaphragmatic breathing for a period of 3 minutes. This will test whether the program can pick up on these moments and therefore test the accuracy and thus validity of the program.

Due to the lengthy procedure of data collection and scheduling issues with experienced breathers, only three breathers were involved with data collection and only one recording session was organized for each of them.

7.3.1 Data Collection Protocol

The participants are requested to use the restroom before the data collection begins and have been asked not to consume a meal 1 hour before the appointment. This prevents their stomachs from bloating, affecting the cross-sectional area of the diaphragm, and therefore the data. The participants are asked to read and sign the Informed Consent Form (Appendix E). The participants are then asked to only wear a single layer of clothing on their torso.

Originally the thoracic band was to be placed at the level of the nipples (either above or below them), however from personal experience with recording data, the author found that the band slips a lot more compared to if it is placed at the level of the armpits. The protocols for putting on the bands are different for men and women.

Men

Men simply wear the thoracic band at the level of the axilla, slightly above the chest. The abdominal band is placed at the level of the navel.

Women

Women are instructed to go to a separate room where they place the chest band under their bra straps, at the level of the axilla and then put on their top layer. This is done to prevent the bands slipping along the bra straps if they were placed over them. They then return to the room with the researcher to attach the abdominal band.

The protocol for recording the training data is as follows:

1. The participant is instructed to practice diaphragmatic breathing for 1 minute.
2. The participant wears the bands appropriately.
3. The participant is instructed to be seated.
4. The participant performs diaphragmatic breathing for 2 and a half minutes.
5. The Airleviate is connected to the computer and the file is saved.
6. The Airleviate is connected to the user again.
7. The participant performs chest breathing for 2 and a half minutes.
8. The Airleviate is connected to the computer and the file is saved.
9. The Airleviate is connected to the user again.
10. The participant is instructed to stand.
11. The participant performs diaphragmatic breathing for 2 and a half minutes.
12. The Airleviate is connected to the computer and the file is saved.
13. The Airleviate is connected to the user again.
14. The participant performs chest breathing for 2 and a half minutes.
15. The Airleviate is connected to the computer and the file is saved.
16. The Airleviate is connected to the user again.
17. The participant is instructed to walk around the room for 10 minutes while stopping as little as possible.
18. The Airleviate is connected to the computer and the file is saved.

After recording the training data, all the data that was recorded while standing and sitting is consolidated into a file that represents stationary data. The stationary and movement data files will train the movement classifier. Likewise, all the data recorded while breathing diaphragmatically is consolidated in to a DB file and the same is done with the chest breathing data.

The participant is informed that their breathing will be recorded for an hour and are asked to prepare a list of times where they will pay special attention to breathing diaphragmatically. Alarms are set for those times, and when each alarm goes off, the participant will spend three minutes breathing diaphragmatically. The Airleviate is connected to the participant and the recording is started. Whenever an alarm goes off, the author sets off a timer for three minutes and the participant is told to breathe diaphragmatically. After an hour, the recording is finished and the author enters the data file into the program.

7.3.2 Results

Three participants were recruited who had all received diaphragmatic breathing training before this project from the same trainer, the supervisor of this report, Erik Faber. This was done to ensure that they were taught in the same way and had learned the same style of diaphragmatic breathing. Three DBMs of 3 minutes each were scheduled during the breathing session, as explained in above.

There are four scenarios that scheduled DBMs can be detected in:

1. The first scenario is where the participant was already in a DBM and therefore the planned DBM was contained in a longer DBM encompassing it.
2. The second scenario was where the planned DBM was detected however was split up into two smaller DBMs.
3. The third scenario was where the first minute of the scheduled DBM was not listed in features however the second and third minute did form a DBM.
4. The fourth scenario is that the scheduled DBM is not detected.

For all three participants, the program registered when they were breathing diaphragmatically. In the list of DBMs in the output file, there were more DBMs identified than just the three that were scheduled with the timers for some of the participants.

An effort was made to pinch the thoracic strap using a clip in order to tighten it and prevent slippage. However, during the recording sessions, all participants experienced notable slippage of the thoracic band. The participants were instructed to quickly adjust the thoracic band if it slipped too far from its original position, however this adjustment did not guarantee the error was removed.

For one of the participants, the program picked up on three diaphragmatic periods, each three minutes in length. However the time stamps did not reflect the times that were decided upon beforehand. For the other two participants, their diaphragmatic breathing moments were picked up and some coincided within a minute of the scheduled time. Upon further inspection of the raw data, it was seen that the slippage had caused the range of values recorded to shift, thus explaining the strange classification results. The effects of slippage are discussed further below in the Section 7.3.3.

It should be noted that the breathers had not been practicing diaphragmatic breathing regularly prior to the experiment and had a short recap session with the researcher before their breathing was recorded for the hour. Therefore their performance may have been suboptimal, during the scheduled DBMs in the breathing recording session, which is another explanation for scenario 2 and 3.

7.3.3 Results Discussion

The occurrence of the first scenario could be reduced by setting a higher tolerance level at the beginning of the program. This would make sure that minutes with less diaphragmatic breathing get removed, and only minutes with a higher percentage of diaphragmatic breathing labeled samples get selected i.e. those that were done with the intention of breathing diaphragmatically.

The second and third scenarios can be explained by the fact that the breathers were not proficient with diaphragmatic breathing. Care was taken to ensure diaphragmatic breathing was being performed during the recording of the training data. The author paid special attention to the chest and diaphragm movement of the participants during the recording of the training data, to ensure the abdomen expanded during inhalation in the case of diaphragmatic breathing, and the chest expanded during inhalation in the case of chest breathing.

Another explanation for the second scenario provided by one of the participants was that during the scheduled DBM, the participant transitioned from diaphragmatic to chest and back to diaphragmatic breathing by mistake.

The third scenario is also explained by the fact that the participants needed time to consciously switch between diaphragmatic and chest breathing, which explains why in some cases, the first minute of the scheduled DBM was not detected. The participants also mentioned that they felt aware of their efforts when changing breathing styles, which made the process more difficult.

The fourth scenario is difficult to explain as there are multiple causes. The first is that the participant could have employed the wrong breathing during the scheduled moments. The second reason is that there was something wrong with the measurement Airleviate itself. After consulting with the client and designer of the Airleviate, he mentioned that the RIP values can be affected by the voltage of the battery and the temperature in the room. He also mentioned that this prototype had many aspects that still needed to be tested and evaluated. Therefore the fourth scenario is difficult to explain, however it can be said that since the program identified parts of scheduled DBMs for the other participants and the code was tested; the possibility that the program is at fault is lower.

The undetected and partially detected DBMs can largely be explained by the slippage of the thoracic band. The slippage of the thoracic band experienced caused a different cross section of the participants chest compartment to be measured (a larger cross sectional area). These values fell outside the range of values that the classifier was trained on, leading to misclassification. Although care was taken to secure the straps by tightening them using clothes pegs, this did not prevent slippage adequately enough.

The cause of the slippage of the thoracic chest was a combination of the convex shape of the chest compartment, as well as the downward and slightly inward slope of the Latissimus Dorsi muscle shown below in Image 7.10 . This combination of factors caused the thoracic band to slip lower at the back first, and as a result slowly pull down the band at the front of the chest. The elastic material that the bands are made out of, is not tight enough to counteract the effects of gravity and slight movement.

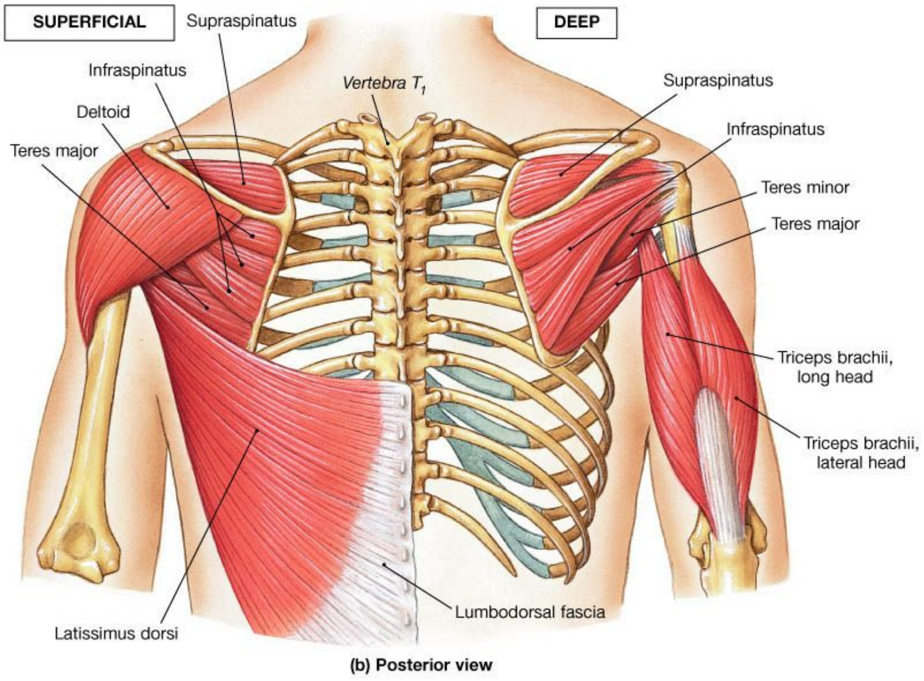


Image 7.10: *Latissimus Dorsi location in the back (Droualb.faculty.mjc.edu, 2018)*

7.4 Functional Test

The functional evaluation as mentioned in Section 3.10, will be testing the functionality described in Revised Requirements, Section 5.6 with the FR tags. The tests will be binary, whether or not the specific requirement has been implemented successfully in the program. Although a binary evaluation is somewhat simplistic, the requirements are specified to a degree to which a yes or no is sufficient to evaluate them. This style of testing is known as unit testing, and tests single functions in a system as was done in the code tests in Section 7.1. Due to many functions simply retrieving or setting variables to a certain value, their functionality is implicitly tested while writing the program. The program would not be able to run without these functions working properly. Some requirements have been italicized signifying a change from findings in Section 6.

Number	**Requirement**	**Check**
	Must Have	
1	Use *the respective contribution from each band to the sum, the raw values and the filtered values* in order to classify the data as “Diaphragmatic” or “Chest” breathing. (FR)	**X**
2	Data classified as “Moving” or “Stationary” for the restful moments using accelerometer data. (FR)	**X**
3	Classify breathing as “Periodic” or “Non-Periodic” to describe periodic breathing using the results of a periodogram. (FR)	**X**
4	Filter breathing data to exclude everything outside the 0.2-0.7 Hz range. (NFR)	**X**
5	Separate data by the minute using a sliding window of 1 minute. (NFR)	**X**
6	Export the data as a text file to send to the interface. (NFR)	**X**
7	User can enter the threshold and buffer length values when the program starts up. (NFR)	**X**

8	Include safeguards to ensure proper parameters entered to run the script. (FR)	**X**
9	Use multiple means of classification and compare the results of each to find the optimal one. (NFR)	**X**
10	Include the following features in the feedback: (NFR) a. Total recording time of the session. b. Calculate a quality factor based off diaphragmatic breathing and repetitive breathing throughout the recorded session. c. Total time spent breathing diaphragmatically. d. Count the number of diaphragmatic breathing periods. e. The longest DBM. f. Times of the longest DBM. g. The average length of a DBM for that session. h. Provide time stamps associated with diaphragmatic breathing moments (DBM) to allow users to back-track and reflect on their activities. i. Include the average contribution of the diaphragm during DBMs. j. The average breathing frequency per minute for the feedback.	**X**
	Should Have	
10	Include the following features for feedback: (NFR) k. The longest repetitive breathing moment. l. The times of the longest repetitive breathing moment. m. Calculate total duration of time spent breathing repetitively. n. List the repetitive breathing moments along with their times and the duration of the moment. o. The number of repetitive breathing periods during the recording sessions	
	Could Have	
10	Include the following features for feedback: (NFR) p. The longest restful moment. q.The times of the longest restful moment. r. Calculate total duration of time spent stationary. s. List the restful moments along with their times and the duration of the moment. t. The number of restful moments during the recording session	
11	Export the features in JSON format (NFR)	**X**

Table 7.1: *Table of requirements that have and have not been met during the realization phase*

Frequencies outside the 0.2 - 0.7 Hz range are filtered and tested in Section 7.1.1. Separating the data using a sliding window of 1 minute is also tested in Section 7.1.2. Exporting the data to text file has been tested implicitly as it was necessary for the collaboration with Florian Naumilkat's project. The threshold and buffer length input has been tested in Section 7.1.7. Multiple means of classification were used and their prediction accuracies compared in Section 6.2. The features being exported were also tested implicitly while collaborating with Florian Naumilkat. Finally the exporting features in JSON format was tested with Florian Naumilkat, but not implemented due to the interface having been adapted to reading text files. The requirement however had been met and was therefore ticked off.

7.5 General Discussion

In this section, problems and discussion points regarding measuring breathing with Airleviate and how breathing is classified in the program will be covered.

7.5.1 Measurement

Mr. Bulsink had explained in his interview (Section 4.5.1) that the sampling frequency of the Airleviate was approximately on the lower side of 8 Hz. This has implications on the filtering of the data as the bandpass filter has an input which is the sampling frequency of the signal. This means the filter did not filter only filter all of the noise but possibly some of the frequencies at the edges of the band (0.2 and 0.7). This does not affect the experiment too much considering that 0.2 Hz amounts to 12 breaths per minute and 0.7 Hz amounts to 42 breaths per minute; neither of which could have been present during the data collection as the participants were only seated or standing.

The Airleviate, containing the accelerometer, is worn on the abdominal band in all the recordings made with it. This means that the accelerometer was not able to capture upper body movement, therefore the accelerometer data is not fully descriptive of the user's movement. The user may be twisting the upper part of their torso, or moving their arms around and the accelerometer could not pick up on it. So the motion labels "Stationary" and "Moving" need to be considered from the context of sitting or standing in one place versus walking around.

Something that the program does not account for is the change in the style of diaphragmatic breathing a user experiences. When the user first learns diaphragmatic breathing, their performance and style will be very different to when they have been practicing it for an extended period of time. In the beginning, the user may puff out their abdomen far more than necessary in order to exaggerate the movement to learn it. The program does not contain the functionality to keep the classifiers up to date with the user's diaphragmatic breathing style. When the Airleviate is released on the market, it should have re-calibration routines to keep the classifiers updated on the user's current diaphragmatic breathing style and performance.

The participants mentioned that there was a difference in performing diaphragmatic breathing when they were sitting and standing. When standing, it was easier to expand the abdomen outward while breathing in, however when sitting it was slightly more difficult and abdomen did not expand as far outward. For the calibration routines in future iterations of the program, a distinction should be made between the two.

The participants all mention that they were very conscious of their breathing during the training data recording sessions. Similar to the experience of the author, they felt they were exaggerating the motions while recording the training data and paying explicit attention to it, as opposed to paying less attention to their breathing while recording the breathing session data. This will change in the future when users become more accustomed to breathing diaphragmatically and do not need to exaggerate the motions in order to confirm that they are performing the right type of breathing.

A complaint that many participants had was that the thoracic band slipped a fair bit during the long recording session. This problem can be attributed to the shape of the chest for most participants and how it induces slippage, however for one participant there was a different problem. The participant had an exceptionally long torso, causing the linking cable between the two bands to fully stretch. The cable was unable to link the two bands without pulling the thoracic band downward. There was relatively less slippage experienced with the abdominal band. Slippage can be fixed by fixing the thoracic band in place using skin-friendly adhesive tape or using bandages.

There was also the problem that the thoracic band was slightly too large for some of the participants with narrower chests, which also encouraged slippage even though it was tightened using clothes pegs. There is no real fix for the issue of a participant with a long torso other than getting a longer linking cable. The material of the bands that the current prototype Airleviate can be replaced with something less stretchable.

7.5.2 Classification

A snapshot of the breathing (represented by a pair of raw band values, their filtered values and their contribution to their sum) was provided to the classifier recorded every 8th of a second and a prediction was provided by the classifier. This method identified DBMs, however there are other methods to approach classifying breathing signals. Samples were grouped by the minute they belonged to; within this however, another step could be taken to find the samples belonging to individual breaths (inhalation and exhalation). This may rule out errors due to users pausing between breaths and gives a better view of the continuous breathing signal. The area under the curve of the sum signal for each breath, and the average abdominal contribution to the sum signal during the inhalation and exhalation could be given as an additional inputs to the classifier. Samples would then be grouped by breath within a given minute. A point for future work would be to explore the difference in results between a classifier that looks at individual samples and one that looks at samples in the context of a breath with additional parameters relating to the breath, for example the area under the curve of the sum signal.

Diaphragmatic breathing was classified, as shown in Image 5.5, by sample. The total number of samples labeled “diaphragmatic” were tallied. Another way this could have been done is to only consider adjacent samples that are labeled “diaphragmatic”. This would be a stricter definition of diaphragmatic than the simple tallying method currently implemented.

Repetitive breathing is calculated based on the breathing frequency and does not consider the form of type of breathing of the previous minutes. True repetitive breathing would distinguish between chest breathing and diaphragmatic breathing. Even if the breathing frequency of a minute falls within one standard deviation of the buffer’s breathing frequency, if the type of breathing changes from chest to diaphragmatic or vice versa, then the breathing in that minute is not considered repetitive.

While investigating and comparing the accuracies of various classifiers in Section 6.2, it was found that the SVM classified both breathing data and accelerometer data with the highest accuracy. However each classifier had various parameters that they could be initialized with. For example the number of hidden layers could be declared when initializing the Neural Network. A point for future work would be to investigate how varying different parameters of the classifiers affects their prediction accuracy. It is possible that a higher prediction accuracy can be obtained through this process.

The methodology used in this program is as follows: the classifiers are trained on the respective participant’s training data for whom the breathing session data is being processed. Dr. Kamilaris mentioned the “transfer knowledge” aspect of the program and whether breathing data can be generalized. The author had not attempted to process a participants’ breathing session data with classifiers trained on another’s training data. The author is under the impression that breathing is very personal and although there are definitely some similarities between how participants perform diaphragmatic breathing, they each have their own slight variations that would lower the quality of the processing if used on another participants data. Also, there can be cases where an individual has an abnormal chest to abdominal cross sectional area ratio. In this case, the relative contribution from the abdomen may be slightly less compared to a user with a close to 1:1 ratio. This is a point for future work.

Instead of appending labels to the end of each sample, a dictionary could have been used to store all the values. A dictionary in python is a data structure where objects can be addressed via set labels instead of positions in the array, allowing the position to be arbitrary. The labels would look like: “R1_value”, “R2_value”, “X_axis” etc. This would also make the coding process easier as values would not need to accessed by their position in the list but by their label.

7.6 Third Requirements Iteration

Following the findings from the code tests, the expert validations and the program verification test, as well as tests performed in Section 6, the requirements are redefined for future work. The parts of the features that have been updated are shown in italics. New requirements are added in bold and italics.

Must Have

1. Use *the respective contribution from each band to the sum, the raw RIP values and the filtered RIP values* in order to classify the data as “Diaphragmatic” or “Chest” breathing. (FR)
2. Data classified as “Moving” or “Stationary” for the restful moments using accelerometer data. (FR)
3. Classify breathing as “Periodic” or “Non-Periodic” to describe periodic breathing using the results of a periodogram *and based on the style of breathing*. (FR)
4. Filter breathing data to exclude everything outside the 0.2-0.7 Hz range. (NFR)
5. Separate data by the minute using a sliding window of 1 minute. (NFR)
6. Export the data as a text file to send to the interface. (NFR)
7. User can enter the threshold and buffer length values when the program starts up. (NFR)
8. Include safeguards to ensure proper parameters entered to run the script. (FR)
9. Use multiple means of classification and compare the results of each to find the optimal one. (NFR)
10. Include the following features in the feedback: (NFR)
    a. Total recording time of the session.
    b. Calculate a quality factor based off diaphragmatic breathing and repetitive breathing throughout the recorded session.
    c. Total time spent breathing diaphragmatically.
    d. Count the number of diaphragmatic breathing periods.
    e. The longest DBM.
    f. Times of the longest DBM.
    g. The average length of a DBM for that session.
    h. Provide time stamps associated with diaphragmatic breathing moments (DBM) to allow users to back-track and reflect on their activities.
    i. Include the average contribution of the diaphragm during DBMs.
    j. The average breathing frequency per minute for the feedback.
11. ***Fasten thoracic and abdominal strap to participant using skin-friendly adhesive tape or a bandage. (NFR)***

Should Have

10. Include the following features for feedback: (NFR)
    k. The longest repetitive breathing moment.
    l. The times of the longest repetitive breathing moment.
    m. Calculate total duration of time spent breathing repetitively.

n. List the repetitive breathing moments along with their times and the duration of the moment.
o. The number of repetitive breathing periods during the recording sessions

Could Have

10. Include the following features for feedback: (NFR)
    p. The longest restful moment.
    q. The times of the longest restful moment.
    r. Calculate total duration of time spent stationary.
    s. List the restful moments along with their times and the duration of the moment.
    t. The number of restful moments during the recording session
12. Export features in JSON format (NFR)

Won't Have

13. Airleviatie should support haptic feedback. (FR)
14. Airleviate should use haptic feedback as a reminder and notification system. (FR)

7.7 Evaluation Conclusion

This chapter served to answer the research sub-question:

"*How can the validity of the classification be verified?*"

The program was tested using the Unit Test method where single components of the code are tested. After that experts in the field of data analysis and breathing were approached to comment on the validity of the methodology and approach to classify breathing data. Lastly, the program as a whole was tested on participants with prior diaphragmatic breathing experience.

The Unit Tests were all conducted successfully, and showed that the program was able to handle errors in the data and user input, as well as various scenarios for calculations, such as the rolling over of an hour or day while calculating the difference in time. The experts that were approached validated the system in terms of methodology, the way the data was processed and the approach to classify breathing. They also made some suggestions for future work. The results of the DBM detection test indicates that the program is able to detect DBMs, both scheduled and otherwise if there are no errors introduced while recording data. Care needs to be taken to prevent slippage by fastening the bands in place using a bandage or tape, as tightening them with clothes pegs is not sufficient.

8. Conclusion

This chapter marks the end of the report, and answers all the research sub-questions listed in Section 1.3 in brief, followed by an answer to the main research question.

8.1 Conclusion

This section will begin by answering, in short, the sub-research questions formulated to help answer the main research question. The first four sub-research questions address the topics of the health benefits of diaphragmatic breathing, the use of RIP as a measurement for breathing, the classification of respiratory data and habit formation, for background information. Using this information, and the information gathered by answering the remaining sub-research questions regarding the kinds of features users like to be provided and testing the validity of the developed program, the main research question is answered.

"What are the physiological health benefits of diaphragmatic breathing?"

By doing literature research, it was found that diaphragmatic breathing has many applications as interventions and treatments to conditions such as oxidative stress, anxiety and hypertension. Diaphragmatic breathing needs to be practiced regularly, for periods of longer than 4 weeks to observe sustained and consistent results. Motivation plays a role in making users stick to practice routines.

"Why is RIP used for recording breathing data?"

From literature, it was found that RIP has been validated as method to measure respiration, and that the position of the bands varies between studies, however is generally at the level of the axilla and navel. It was concluded that it is not an all purpose breathing measurement technique and suffers during activities involving large amounts of physical movement and activity. Slippage of the thoracic band was found to be a major source of error as can be confirmed from the results of the Program Verification test.

"How can respiratory data be classified?"

The types of classifiers used for breathing pattern recognition vary between studies. The classifiers used depended on the complexity of the data being recorded and the goal of the classification. The features extracted from the data were loosely dependent on the goal of the studies. Some studied focused more on time domain components while others on those from the frequency domain. Each classifier excelled at performing certain types of classification and one cannot be used to do everything.

"How is a habit formed?"

Habits are formed when an action is repeatedly done as a response to a certain cue or context. The amount of time a habit takes to form varies significantly between individuals and is dependent on their motivation to form the habit and the independence they are given to form said habit. Habits can weaken over time if not performed frequently and is a point to reflect upon when weaning users off the Airleviate, to perform diaphragmatic breathing on their own.

"*What kinds of breathing features would users like to be presented for the purpose of feedback?*"

From the brainstorms, it was found that users are interested in statistics like the a quality factor to describe the performance during a session, the longest DBM and the average DBM duration. Similar features were thought up during the in individual brainstorm, showing that the author and potential users have similar ideas in mind for what the breathing feedback should consist of. The interviews showed that the client had an interest in moments of rest (where the user is relatively stationary) and moments of periodic breathing (where the user's breathing frequency does not vary too much with time). The complete list of features can be found in the Preliminary Requirements in Section 4.8.

"How can the validity of the program be verified?"

The program was tested using the Unit Test method where single components of the code are tested. The Unit Tests were all conducted successfully, and showed that the program was able to handle errors in the data and user input, as well as various scenarios for calculations including the rolling over of an hour or day while calculating the difference in time, or detecting DBMs.

After that experts in the field of data analysis and breathing were approached to comment on the validity of the methodology and approach to breathing classification the program took. The experts approached validated the system in terms of methodology, the way the data was processed and the approach to classify breathing.

Lastly, the program as a whole was tested on participants with prior diaphragmatic breathing experience. The results of the DBM detection test indicate that the program is able to detect DBMs, both scheduled and otherwise if there are no errors introduced while recording data. Care needs to be taken to prevent slippage by fastening the bands in place using a bandage or tape, as tightening them with clothes pegs is not sufficient. The thoracic band is more likely to slip than the abdominal band so extra care should be taken to fasten it.

The main research question of this project was:

“How can Respiratory Inductance Plethysmography data be classified and analyzed to be used as feedback towards cultivating habitual diaphragmatic breathing?”

A program was developed to identify periods of diaphragmatic breathing, and to extract various other features regarding diaphragmatic breathing from respiratory data recorded using RIP. Unwanted noise outside the 0.2 - 0.7 Hz frequency range was filtered using a Bandpass filter. Diaphragmatic breathing is classified using a Support Vector Machine to label samples as “Diaphragmatic” or “Chest” as well as “Moving” or “Stationary”, based on accelerometer data. Repetitive breathing is evaluated using breathing frequency, that is extracted using a periodogram and comparing it to the average breathing frequency of previous minutes. Statistics such as a quality factor for the session, the longest DBM, the average DBM for the session etc. are extracted and presented to the user as feedback towards cultivating habitual diaphragmatic breathing, as these features are of interest. The program was able to identify some DBMs, however errors were introduced due to slippage of the thoracic band. Attention should be paid to the placement and securing of the bands to reduce errors introduced by the slippage of the thoracic or abdominal band.

9. Recommendations for Future Work

This chapter serves to cover topics that were discovered during the course of this project, but which did not fall under the scope of this project as well as requirements that were not fulfilled.

9.1 Recommendations

Implementing the feature extraction for the repetitive and restful moments is the next step in this project. Providing users with a more holistic overview of their activities by comparing breathing to movement and breathing rate will help form a better understanding of their habits and equip them with the knowledge to change said habits. Real time classification would also greatly aid in this, and combined with haptic feedback would provide an even better overview of the wearer's breathing.

The validity of the program should further be tested by fastening the thoracic and abdominal bands in place, and recording more breathing sessions. The bands should be fastened using a bandage or skin-friendly adhesive tape. More DBMs should be scheduled during the "real world" test, and the sessions should last longer than just one hour, in order to simulate real usage. After implementing fastened bands into the experimental protocol, the effects of movement should be tested on the program to see how much the data is affected by movement, and whether or not there is less slippage experienced.

For development of real-time data analysis, the author would recommend using Bluetooth technology to connect the Airleviate to the program. Each message the Airleviate sends should begin with a header with a uniquely identifiable ID for the package as well as an ending to signify the end of the message. The Airleviate and program should also support a confirmation system where the program sends a confirmation that the package has been received and that the Airleviate should send the next package in its buffer. The Airleviate saves the packages to send in a buffer and removes them only after receiving a confirmatory message from the program. This will help to ensure data transfer without packages being missed or lost.

From the Expert Validation with Dr. Kamilaris, it was found the concept of "Transfer Knowledge" was an interesting field to do more research in. This relates to transferring the knowledge the classifier has on a user's breathing and applying it when evaluating other users' breathing. Further research can be done into comparing the prediction accuracy of classifiers trained on public and private diaphragmatic breathing data. In addition, constructing a confusion matrix to see where the classifiers made errors would provide insight into the classification errors. Dr. Kamilaris also recommended exporting the weights the classifier assigns to the various features to investigate which features the classifier finds more important.

With regards to classification, there are several points for future work. The effect and tuning of various classifier parameters when initialized can be evaluated to narrow down on the optimal classifier parameters for the task of classifying diaphragmatic breathing. The classifiers can also be trained on additional features, if the data is looked at in terms of individual breaths within a minute. Comparing the prediction accuracy of classifiers trained on samples in a minute versus samples per breath is an interesting topic and further expands the ways in which diaphragmatic breathing classification can be approached. One such feature can be the sum of the X, Y and Z accelerometer axes as explained in Section 7.2.2. K fold cross validation can be used while training the classifiers to make them more robust and well suited to “real world” scenarios.

10. References

Arden-Close, E., Yardley, L., Kirby, S., Thomas, M., & Bruton, A. (2017). Patients' experiences of breathing retraining for asthma: a qualitative process analysis of participants in the intervention arms of the BREATHE trial. *NPJ Primary Care Respiratory Medicine*, *27*, 56. http://doi.org/10.1038/s41533-017-0055-5

Abraham, W., Watson, H., Schneider, A., King, M., Yerger, L. and Sackner, M. (1981). Noninvasive ventilatory monitoring by respiratory inductive plethysmography in conscious sheep. *Journal of Applied Physiology*, [online] 51(6), pp.1657-1661. Available at: https://www.ncbi.nlm.nih.gov/pubmed/7319896 [Accessed 14 Apr. 2018].

Agile Business Consortium. (2018). DSDM Atern Handbook (2008). [online] Available at: https://www.agilebusiness.org/content/moscow-prioritisation-0 [Accessed 3 May 2018].

App Store. (2018). BellyBio Interactive Breathing on the App Store. [online] Available at: https://itunes.apple.com/us/app/bellybio-interactive-breathing/id353763955?mt=8 [Accessed 23 Apr. 2018].

Apple A. (n.d.). iOS - Health. [online] Available at: https://www.apple.com/ios/health/ [Accessed 27 Jun. 2018].

Apple B. (n.d.). Apple Watch - Close Your Rings. [online] Available at: https://www.apple.com/watch/close-your-rings/ [Accessed 27 Jun. 2018].

Bradley, H., & Dr. Esformes, J. (2014). BREATHING PATTERN DISORDERS AND FUNCTIONAL MOVEMENT. *International Journal of Sports Physical Therapy*, *9*(1), 28–39.

Boxbreathing.org. (2018). Box Breathing App. [online] Available at: http://boxbreathing.org/ [Accessed 23 Apr. 2018].

Bruton, A., Lee, A., Yardley, L., Raftery, J., Arden-Close, E., Kirby, S., … Thomas, M. (2018). Physiotherapy breathing retraining for asthma: a randomised controlled trial. *The Lancet. Respiratory Medicine*, *6*(1), 19–28. http://doi.org/10.1016/S2213-2600(17)30474-5

Brugha, R., & Varvasovszky, Z. (2000). Stakeholder analysis: a review. *Health policy and planning*, 15(3), 239-246. Available at: https://academic.oup.com/heapol/article-abstract/15/3/239/573296 [Accessed 30 Apr. 2018].

Business.gov.nl. (2018). Medical devices. [online] Available at: https://business.gov.nl/regulation/medical-devices/ [Accessed 30 Apr. 2018].

BusinessDictionary.com. (2018). Brainstorming. [online] Available at: http://www.businessdictionary.com/definition/brainstorming.html [Accessed 3 May 2018].

Caretti, D., Pullen, P., Premo, L. and Kuhlmann, W. (1994). Reliability of Respiratory Inductive Plethysmography for Measuring Tidal Volume During Exercise. *American Industrial Hygiene Association Journal*, [online] 55(10), pp.918-923. Available at: http://www.dtic.mil/dtic/tr/fulltext/u2/a266883.pdf [Accessed 5 Apr. 2018].

Cohen, K., Ladd, W., Beams, D., Sheers, W., Radwin, R., Tompkins, W. and Webster, J. (1997). Comparison of impedance and inductance ventilation sensors on adults during breathing, motion, and simulated airway obstruction. IEEE Transactions on Biomedical Engineering, 44(7), pp.555-566.

Chadha, T., Watson, H., Birch, S., Jenouri, G., Schneider, A., Cohn, M. and Sackner, M. (1982). Validation of Respiratory Inductive Plethysmography Using Different Calibration Procedures. *American Review of Respiratory Disease*, [online] 125(6), pp.644-649. Available at: https://www.atsjournals.org/doi/abs/10.1164/arrd.1982.125.6.644 [Accessed 13 Apr. 2018].

Chris V. Nicholson, S. (2018). Introduction to Deep Neural Networks (Deep Learning) - Deeplearning4j: Open-source, Distributed Deep Learning for the JVM. Retrieved from https://deeplearning4j.org/neuralnet-overview

Chang, S., Kim, H., Ko, Y., Bae, C. and An, S. (2009). Effects of Abdominal Breathing on Anxiety, Blood Pressure, Peripheral Skin Temperature and Saturation Oxygen of Pregnant Women in Preterm Labor. *Korean Journal of Women Health Nursing*, [online] 15(1), 32. Available at: https://synapse.koreamed.org/search.php?where=aview&id=10.4069/kjwhn.2009.15.1.32&code=0102KJWHN&vmode=PUBREADER [Accessed 20 Mar. 2018].

Courtney, R., van Dixhoorn, J. and Cohen, M. (2008). Evaluation of Breathing Pattern: Comparison of a Manual Assessment of Respiratory Motion (MARM) and Respiratory Induction Plethysmography. *Applied Psychophysiology and Biofeedback*, [online] 33(2), pp.91-100. Available at: https://fysio-posthoes.nl/dynamic/media/26/documents/MARM_validering%20en%20betrouwbaarheid.pdf [Accessed 5 Apr. 2018].

Chem-eng.utoronto.ca. (2018). LDA. [online] Available at: http://chem-eng.utoronto.ca/~datamining/dmc/lda.htm [Accessed 23 Apr. 2018].

Chen, Y., Huang, X., Chien, C., and Cheng, J. (2016). The Effectiveness of Diaphragmatic Breathing Relaxation Training for Reducing Anxiety. *Perspectives In Psychiatric Care*, 53(4), 329-336. doi:10.1111/ppc.12184

Courtney, R. (2017). Breathing training for dysfunctional breathing in asthma: taking a multidimensional approach. *ERJ Open Research*, *3*(4), 00065–2017. http://doi.org/10.1183/23120541.00065-2017

Courtney, R., van Dixhoorn, J. and Cohen, M. (2008). Evaluation of Breathing Pattern: Comparison of a Manual Assessment of Respiratory Motion (MARM) and Respiratory Induction Plethysmography. *Applied Psychophysiology and Biofeedback*, [online] 33(2), pp.91-100. Available at: https://fysio-posthoes.nl/dynamic/media/26/documents/MARM_validering%20en%20betrouwbaarheid.pdf [Accessed 5 Apr. 2018].

David, R. (n.d.). Types of Brainstorming. [online] Brainstorming- instructional strategy. Available at: https://brainstorm-rebeccadavid.weebly.com/types-of-brainstorming.html [Accessed 27 Jun. 2018].

Depiazzi, J. and Everard, M. (2016). Dysfunctional breathing and reaching one's physiological limit as causes of exercise-induced dyspnoea. *Breathe*, [online] 12(2), pp.120-129. Available at: http://breathe.ersjournals.com/content/12/2/120#sec-7 [Accessed 10 Apr. 2018].

Droualb.faculty.mjc.edu. (2018). Muscle Identification. [online] Available at: http://droualb.faculty.mjc.edu/Lecture%20Notes/Unit%203/muscles%20with%20figures.htm [Accessed 30 Jun. 2018].

Envelopes. (n.d.). 1st ed. [ebook] Auburn: Auburn University, p.62. Available at: http://www.eng.auburn.edu/~troppel/courses/TIMS-manuals-r5/TIMS%20Experiment%20Manuals/Student_Text/Vol-A1/a1-05.pdf [Accessed 20 Apr. 2018].

Fleming, S. and Tarassenko, L. (2007). A Comparison of Signal Processing Techniques for the Extraction of Breathing Rate from the Photoplethysmogram. International Journal of Biological and Medical Sciences, [online] 2(4), p.233-234. Available at: https://www.researchgate.net/profile/L_Tarassenko/publication/228907148_A_Comparison_of_signal_processing_techniques_for_the_extraction_of_breathing_rate_from_the_photoplethysmogram/links/00463521ce4cc3ca4b000000.pdf [Accessed 17 Jun. 2018].

Fitbit.com. (n.d.). Fitbit Official Site for Activity Trackers and More. [online] Available at: https://www.fitbit.com/nl/home [Accessed 27 Jun. 2018].

FocusBand. (2018). How do the FocusBand works for brain training and in-motion. [online] Available at: https://focusband.com/how-focusband-works/ [Accessed 14 Apr. 2018].

FocusBand. (2018). What is the NeuroSelf care app? [online] Available at: https://focusband.com/neuroself-care [Accessed 14 Apr. 2018].

Freeletics.com. (n.d.). FREELETICS - functional high intensity bodyweight training. [online] Available at: https://www.freeletics.com/en [Accessed 27 Jun. 2018].

Garde, A., Sörnmo, L., Jané, R. and Giraldo, B. (2010). Breathing Pattern Characterization in Chronic Heart Failure Patients Using the Respiratory Flow Signal. *Annals of Biomedical Engineering*, [online] 38(12), pp.3572-3580. Available at: https://link.springer.com/article/10.1007%2Fs10439-010-0109-0 [Accessed 5 Apr. 2018].

Garde, A., Giraldo, B., Jane, R., Diaz, I., Herrera, S., Benito, S., Domingo, M. and Bayes-Genis, A. (2008). Characterization of periodic and non-periodic breathing pattern in chronic heart failure patients. *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. [online] Available at: http://ieeexplore.ieee.org/document/4649891/ [Accessed 5 Apr. 2018].

Gardner, B. (2012). Habit as automaticity, not frequency. *European Health Psychologist*, [online] 14. Pp.32-36. Available at: https://www.researchgate.net/profile/Benjamin_Gardner/publication/230576965_Habit_as_automaticity_not_frequency/links/09e4150753b94afa13000000/Habit-as-automaticity-not-frequency.pdf [Accessed 14 Apr. 2018]

Gardner, B., Lally, P. and Wardle, J. (2012). Making health habitual: the psychology of 'habit-formation' and general practice. *British Journal of General Practice,* [online] 62(605), pp.664-666. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3505409/ [Accessed 14 Apr. 2018].

Gardner, B., Sheals, K., Wardle, J. and McGowan, L. (2014). Putting habit into practice, and practice into habit: a process evaluation and exploration of the acceptability of a habit-based dietary behaviour change intervention. *International Journal of Behavioral Nutrition and Physical Activity*, [online] 11(1). Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4216661/ [Accessed 14 Apr. 2018].

Girodo M, Ekstrand KA and Metivier GJ (1992). Deep diaphragmatic breathing: rehabilitation exercises for the asthmatic patient. *Arch Phys Med Rehabil*, 73, 717-20. Available at: http://www.archives-pmr.org/article/0003-9993(92)90204-A/pdf [Accessed 20 Mar. 2018].

Grammatopoulou, E., Skordilis, E., Stavrou, N., Myrianthefs, P., Karteroliotis, K., Baltopoulos, G. and Koutsouki, D. (2011). The Effect of Physiotherapy-Based Breathing Retraining on Asthma Control. *Journal of Asthma*, 48(6), pp.593-601.

Heart.org. (2018). Understanding Blood Pressure Readings. [online] Available at: http://www.heart.org/HEARTORG/Conditions/HighBloodPressure/KnowYourNumbers/Understanding-Blood-Pressure-Readings_UCM_301764_Article.jsp [Accessed 19 Apr. 2018].

Hegde, S., Adhikari, P., Subbalakshmi, N., Nandini, M., Rao, G., & D'Souza, V. (2012). Diaphragmatic breathing exercise as a therapeutic intervention for control of oxidative stress in type 2 diabetes mellitus. *Complementary Therapies In Clinical Practice*, 18(3), 151-153. doi: 10.1016/j.ctcp.2012.04.002

Jones, M., Harvey, A., Marston, L. and O'Connell, N. (2013). Breathing exercises for dysfunctional breathing/hyperventilation syndrome in adults. *Cochrane Database of Systematic Reviews*. [online] Available at: http://cochranelibrary-wiley.com/doi/10.1002/14651858.CD009041.pub2/abstract [Accessed 18 Apr. 2018].

Judah, G., Gardner, B. and Aunger, R. (2012). Forming a flossing habit: An exploratory study of the psychological determinants of habit formation. British Journal of Health Psychology, 18(2), pp.338-353.

Kaushik, R., Kaushik, R., Mahajan, S. and Rajesh, V. (2005). Biofeedback assisted diaphragmatic breathing and systematic relaxation versus propranolol in long term prophylaxis of migraine. *Complementary Therapies in Medicine*, [online] 13(3), pp.165-174. Available at: https://www.sciencedirect.com/science/article/pii/S0965229905000439 [Accessed 14 Apr. 2018].

Kelley, J. (1984). An iterative design methodology for user-friendly natural language office information applications. [online] cs.uwaterloo.ca. Available at: https://www.student.cs.uwaterloo.ca/~cs449/s17/p26-kelley.pdf [Accessed 24 June. 2018].

Kuspa, K., & Pratkanis, T. (2013). Classification of Mobile Device Accelerometer Data for Unique Activity Identification [Ebook]. Palo Alto. Retrieved from http://cs229.stanford.edu/proj2013/PratkanisKuspa-ClassificationOfMobileDeviceAccelerometerDataforUniqueActivityIdentification.pdf

Lally, P., Van Jaarsveld, C., Potts, H. and Wardle, J. (2009). How are habits formed: Modelling habit formation in the real world. *European Journal of Social Psychology*, [online] 40(6), pp.998-1009. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/ejsp.674 [Accessed 14 Apr. 2018].

Larburu, N., Widya, I., Bults, R., Hermens, H. and Napolitano, C. (2013). Early phase telemedicine requirements elicitation in collaboration with medical practitioners. 2013 21st IEEE International Requirements Engineering Conference (RE). [online] Available at: https://ieeexplore.ieee.org/document/6636729/ [Accessed 27 Jun. 2018].

Lee, S., Motai, Y., Weiss, E. and Sun, S. (2012). Irregular Breathing Classification From Multiple Patient Datasets Using Neural Networks. *IEEE Transactions on Information Technology in Biomedicine*, [online] 16(6), pp.1253-1264. Available at: http://ieeexplore.ieee.org/abstract/document/6276258/?reload=true [Accessed 4 Apr. 2018].

Li and Yeh (2005). The application of Qigong therapy to health care. *J. Nurs*, [online] 52, pp.65–70. Available at: https://www.ncbi.nlm.nih.gov/pubmed/15986304 [Accessed 16 Apr. 2018]

Lopez de Mantaras, R. and Armengol, E. (1998). Machine learning from examples: Inductive and Lazy methods. *Data & Knowledge Engineering*, 25(1-2), pp.99-123.

Ma, X., Yue, Z.-Q., Gong, Z.-Q., Zhang, H., Duan, N.-Y., Shi, Y.-T., … Li, Y.-F. (2017). The Effect of Diaphragmatic Breathing on Attention, Negative Affect and Stress in Healthy Adults. *Frontiers in Psychology*, 8, 874. http://doi.org/10.3389/fpsyg.2017.00874

Mader, A. H., & Eggink, W. (2014). A Design Process for Creative Technology. In E. Bohemia, A. Eger, W. Eggink, A. Kovacevic, B. Parkinson, & W. Wits (Eds.), Proceedings of the 16th International conference on Engineering and Product Design, E&PDE 2014 (pp. 568-573). (E&PDE). Bristol, UK: The Design Society.

Mathworks.com. (2018). What is MATLAB?. [online] Available at: https://www.mathworks.com/discovery/what-is-matlab.html [Accessed 30 Apr. 2018].

Martarelli, D., Cocchioni, M., Scuri, S., & Pompei, P. (2011a). Diaphragmatic Breathing Reduces Postprandial Oxidative Stress. *The Journal Of Alternative And Complementary Medicine*, 17(7), 623-628. http://dx.doi.org/10.1089/acm.2010.0666

Martarelli, D., Cocchioni, M., Scuri, S., & Pompei, P. (2011b). Diaphragmatic Breathing Reduces Exercise-Induced Oxidative Stress. *Evidence-Based Complementary And Alternative Medicine*, 2011, 1-10. doi: 10.1093/ecam/nep169

Merriam-webster.com. (2018). Medical Definition of OXIDATIVE STRESS. [online] Available at: https://www.merriam-webster.com/medical/oxidative%20stress [Accessed 19 Apr. 2018].

Merriam-webster.com. (2018). Definition of ANXIETY. [online] Available at: https://www.merriam-webster.com/dictionary/anxiety [Accessed 19 Apr. 2018].

Merriam-webster.com. (2018). Definition of BLOOD PRESSURE. [online] Available at: https://www.merriam-webster.com/dictionary/blood%20pressure [Accessed 19 Apr. 2018].

Murphy, D., Renninger, J. and Schramek, D. (2010). Respiratory inductive plethysmography as a method for measuring ventilatory parameters in conscious, unrestrained dogs. *Journal of Pharmacological and Toxicological Methods*, [online] 62(1), pp.47-53. Available at: https://www.ncbi.nlm.nih.gov/pubmed/20435149 [Accessed 10 Apr. 2018].

WEESE-MAYER, D., CORWIN, M., PEUCKER, M., DI FIORE, J., HUFFORD, D., TINSLEY, L., NEUMAN, M., MARTIN, R., BROOKS, L., DAVIDSON WARD, S., LISTER, G. and WILLINGER, M. (2000). Comparison of Apnea Identified by Respiratory Inductance Plethysmography with That Detected by End-tidal CO2or Thermistor. *American Journal of Respiratory and Critical Care Medicine*, [online] 162(2), pp.471-480. Available at: http://www.atsjournals.org/doi/abs/10.1164/ajrccm.162.2.9904029 [Accessed 5 Apr. 2018].

nlp.stanford.edu. (n.d.). Linear versus nonlinear classifiers. [online] Available at: https://nlp.stanford.edu/IR-book/html/htmledition/linear-versus-nonlinear-classifiers-1.html [Accessed 20 Apr. 2018].

Pages.cs.wisc.edu. (2018). A Basic Introduction To Neural Networks. [online] Available at: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html [Accessed 23 Apr. 2018].

Pollard, D. (1997). *Statistics 241* [Ebook] (p. 1). Connecticut: Yale. Retrieved from http://www.stat.yale.edu/~pollard/Courses/241.fall97/Variance.pdf

Prasertsung, P., & Horanont, T. (2016). A classification of accelerometer data to differentiate pedestrian state. 2016 International Computer Science And Engineering Conference (ICSEC). doi: 10.1109/icsec.2016.7859881

Prana.co. (2018). Prana Tech. [online] Available at: http://prana.co/science [Accessed 14 Apr. 2018].

Python.org. (2018). What is Python? Executive Summary. [online] Available at: https://www.python.org/doc/essays/blurb/ [Accessed 30 Apr. 2018].

Rai, P. (2011). Kernel Methods and Nonlinear Classification.

Raoufy, M., Ghafari, T., Darooei, R., Nazari, M., Mahdaviani, S., Eslaminejad, A., Almasnia, M., Gharibzadeh, S., Mani, A. and Hajizadeh, S. (2016). Classification of Asthma Based on Nonlinear Analysis of Breathing Pattern. *PLOS ONE*, [online] 11(1), p.e0147976. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4732950/ [Accessed 12 Apr. 2018].

Raoufy, M., HAJIZADEH, S., GHARIBZADEH, S., MANI, A., EFTEKHARI, P. and MASJEDI, M. (2012). Nonlinear model for estimating respiratory volume based on thoracoabdominal breathing movements. *Respirology*, [online] 18(1), pp.108-116. Available at: https://www.ncbi.nlm.nih.gov/pubmed/22897148 [Accessed 12 Apr. 2018].

Ravi, N., Dandekar, N. (2005). Activity recognition from accelerometer data. *Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence,* [online] pp.1541-1546. Available at: http://www.aaai.org/Papers/AAAI/2005/IAAI05-013.pdf [Accessed 14 May 2018]

Retory, Y., Niedzialkowski, P., de Picciotto, C., Bonay, M. and Petitjean, M. (2016). New Respiratory Inductive Plethysmography (RIP) Method for Evaluating Ventilatory Adaptation during Mild Physical Activities. *PLOS ONE*, [online] 11(3), p.e0151983. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4805261/#!po=48.1818 [Accessed 12 Apr. 2018].

Rose, K. (1994). Unstructured and semi-structured interviewing. Nurse Researcher, [online] 1(3), pp.23-30. Available at: https://journals.rcni.com/doi/10.7748/nr.1.3.23.s4 [Accessed 1 May 2018].

Reinius, J. (2011). The PACT Analysis Framework. A case study of 1177.se (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:bth-3361

Runtastic.com. (n.d.). Runtastic: Running, Cycling & Fitness GPS Tracker. [online] Available at: https://www.runtastic.com [Accessed 27 Jun. 2018].

Scikit-learn.org. (2018). scikit-learn: machine learning in Python — scikit-learn 0.19.1 documentation. [online] Available at: http://scikit-learn.org/stable/index.html [Accessed 30 Apr. 2018].

Schoonjans, F. (2018). Logistic regression. [online] MedCalc. Available at: https://www.medcalc.org/manual/logistic_regression.php [Accessed 23 Apr. 2018].

Spire. (2018). Science of Breathing. [online] Available at: https://spire.io/pages/science [Accessed 14 Apr. 2018].

Sensors for Wearable Systems. (2010). Wearable Monitoring Systems, [online] pp.3-25. Available at: https://www.researchgate.net/publication/278711880_Sensors_for_Wearable_Systems [Accessed 13 Apr. 2018].

Sharp, H., Finkelstein, A., and Galal, G., (1999). Stakeholder identification in the requirements engineering process. *Proceedings. Tenth Int. Work. Database Expert Syst. Appl. DEXA 99*, [online] pp. 1–5. Available at: http://discovery.ucl.ac.uk/744/1/1.7_stake.pdf [Accessed 30 Apr. 2018]

Singh, A., Thakur N., and Sharma A (2016). A review of supervised machine learning algorithms. *3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, New Delhi, pp. 1310-1315.

Sqa.org.uk. (2007). Functional and Non-Functional Requirements. [online] Available at: https://www.sqa.org.uk/e-learning/SDM03CD/page_02.htm [Accessed 27 Jun. 2018].

TensorFlow. (2018). TensorFlow. [online] Available at: https://www.tensorflow.org/ [Accessed 30 Apr. 2018].

Thomas, M., McKinley, R., Freeman, E., Foy, C., Prodger, P. and Price, D. (2003). Breathing retraining for dysfunctional breathing in asthma: a randomised controlled trial. *Thorax*, 58(2), pp.110-115.

Tobin, M., Jenouri, G., Lind, B., Watson, H., Schneider, A. and Sackner, M. (1983). Validation of Respiratory Inductive Plethysmography in Patients with Pulmonary Disease. *Chest*, [online] 83(4), pp.615-620. Available at: http://journal.chestnet.org/article/S0012-3692(15)41888-4/fulltext#cesec30 [Accessed 14 Apr. 2018].

Tobias, R. (2009). Changing behavior by memory aids: A social psychological model of prospective memory and habit development tested with dynamic field data. Psychological Review, 116(2), pp.408-438.

Topapps.net. (2018). Breathe2Relax – Top Apps. [online] Available at: http://www.topapps.net/apple-ios/top-free-medical-apps-for-iphone.html/attachment/breathe2relax/ [Accessed 23 Apr. 2018].

Tolahunase, M., Sagar, R. and Dada, R. (2017). Erratum to "Impact of Yoga and Meditation on Cellular Aging in Apparently Healthy Individuals: A Prospective, Open-Label Single-Arm Exploratory Study". *Oxidative Medicine and Cellular Longevity*, [online] 2017, pp.1-7. Available at: https://www.researchgate.net/publication/312508689_Impact_of_Yoga_and_Meditation_on_Cellular_Aging_in_Apparently_Healthy_Individuals_A_Prospective_Open-Label_Single-Arm_Exploratory_Study [Accessed 16 Apr. 2018].

Wang, S., Li, S., Xu, X., Lin, G., Shao, L., Zhao, Y. and Wang, T. (2010). Effect of Slow Abdominal Breathing Combined with Biofeedback on Blood Pressure and Heart Rate Variability in Prehypertension. *The Journal of Alternative and Complementary Medicine*, 16(10), pp.1039-1045.

Wang, L. (2005). Support Vector Machines: Theory and Applications. [ebook] Berlin: Springer, p.2. Available at: http://www.ntu.edu.sg/home/elpwang/PDF_web/05_SVM_basic.pdf [Accessed 23 Apr. 2018].

Wilson, C. (2014). User interface inspection methods. Massachusetts: Elsevier, pp.65-79.

YouTube. (2018). Bellybio interactive | Health-e-Apps | Improve Your Health In A Mobile Minute. [online] Available at: https://www.youtube.com/watch?v=e5GD5a55PWM [Accessed 14 Apr. 2018].

YouTube. (2018). Breathe2Relax - Overview. [online] Available at: https://www.youtube.com/watch?v=AEZ1dvZuJYc&list=PL8ABF9C2342A85A0A [Accessed 14 Apr. 2018].

Yu, W. and Song, J. (2010). Effects of Abdominal Breathing on State Anxiety, Stress, and Tocolytic Dosage for Pregnant Women in Preterm Labor. *Journal of Korean Academy of Nursing*, [online] 40(3), p.442. Available at: https://www.ncbi.nlm.nih.gov/pubmed/20634635 [Accessed 20 Mar. 2018].

11. Appendix

11 A. Expert Interview Questions

11 A.1 Interview with Ineke Ter Hedde

Below are the questions posed to Mrs. Ter Hedde during the interview in Section 2.2.2.

What are the backgrounds of the people that come to you?

Why do people come to you for training?

Do they come with any complaints?

Which areas of the body do your exercises target?

How long do you tell your patients to practice the breathing exercises typically?

How long does it usually take for a patient doing belly breathing exercises to recognize first positive effects?

Do you ever suggest breathing exercises working with the abdomen/belly?

What are the reports of the effects of belly/abdominal breathing that you have heard?

What is your experience with belly/abdominal breathing in your field and in general?

11 A.2 Interview with Parviz Sassanian

The interview questions posed to Mr. Sassanian in Section 2.2.1 are listed below:

Which kinds of people come to your practice/which fields do they work in?

What are the different reasons you would tell your clients to start or increase the time they spend belly/abdominal breathing?

How long does it usually take for a patient doing belly breathing exercises to recognize first positive effects?

What have you found/learned/experienced the benefits of belly/abdominal breathing are?

How often should people perform diaphragmatic breathing every day and for how long?

What kinds of issues can belly/abdominal breathing generally help with?

Are there some problems that can be completely solved with belly/abdominal breathing or is more a part of the remedy?

Do you ever ask your clients to perform belly/abdominal breathing aside from when they are meditating or always only during the activity?

Do your clients ever use apps to help train themselves or practice their exercises?

How do you teach your clients belly/abdominal breathing? Do you use different methods and if so then why?

Do you believe it should be done with a goal in mind?

11 A.3 Interview with Ainara Martinez Garde

The interview questions posed to Mrs. Garde in Section 2.2.3 are listed below:

How did you measure the breathing of the patients?

How invasive did they find that measurement method?

Which other measurement techniques did you consider and why did you not go with them in the end?

What features/characteristics of the signals were important for data analysis that you extracted for the classification?

Can you recommend any sources regarding classification of breathing patterns?

Did you encounter any difficulties while measuring the breathing?

Is there literature about this or did you find them through trial and error?

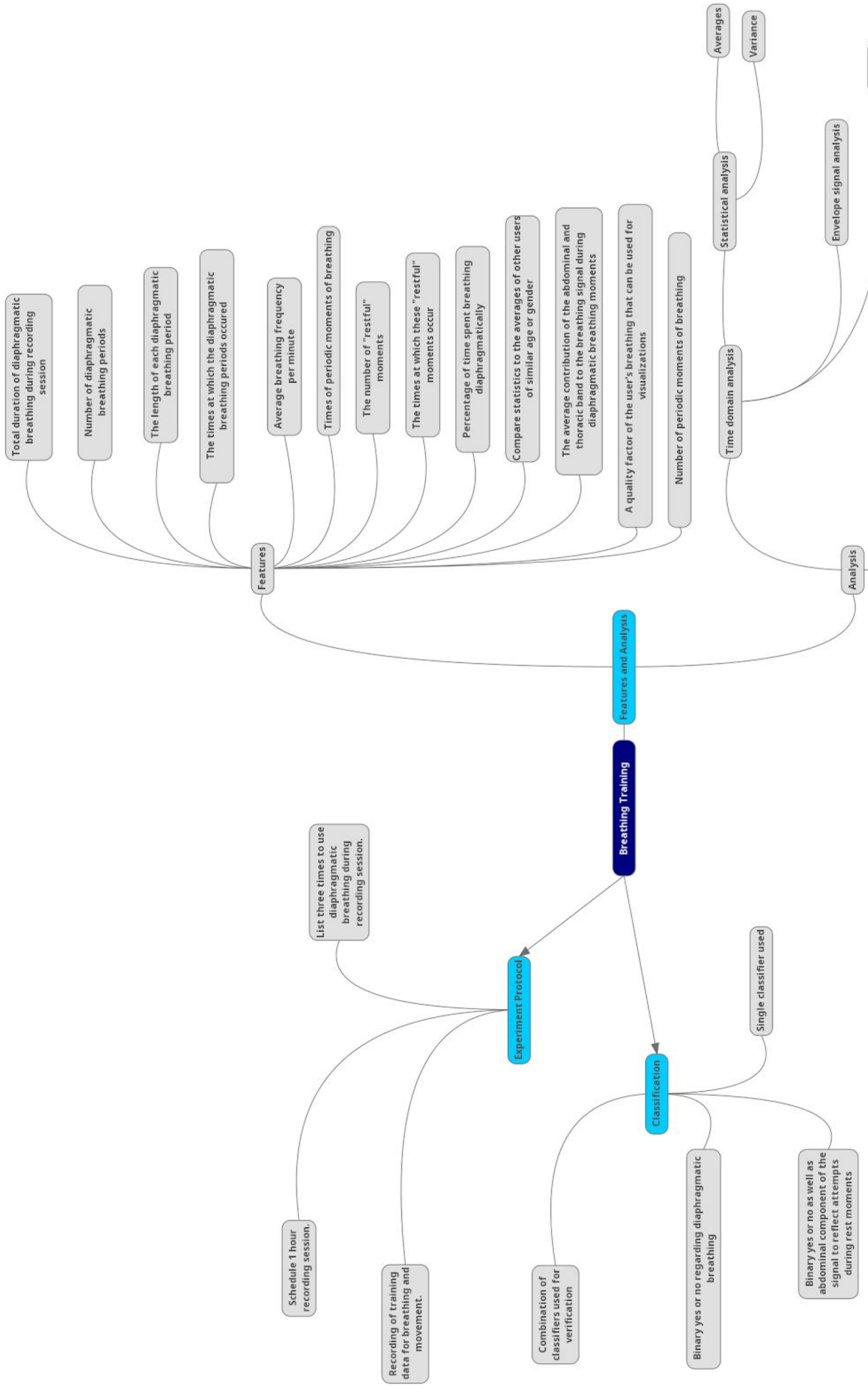
Did you need to approach the ethical committee for your chosen method of breathing measurement?

Where is the processing of the data done? (on the phone or sent to a server)

What type of processing do you do/What kind of AI was used for the classification? (Neural network or another method)

11 B. Individual Brainstorm

Below, a visualization of the individual brainstorm conducted in Section 4.2 can be found in the form of a mindmap:


Breathing Training
Features and Analysis
Features
Total duration of diaphragmatic breathing during recording session
Number of diaphragmatic breathing periods
The length of each diaphragmatic breathing period
The times at which the diaphragmatic breathing periods occured
Average breathing frequency per minute
Times of periodic moments of breathing
The number of "restful" moments
The times at which these "restful" moments occur
Percentage of time spent breathing diaphragmatically
Compare statistics to the averages of other users of similar age or gender
The average contribution of the abdominal and thoracic band to the breathing signal during diaphragmatic breathing moments
A quality factor of the user's breathing that can be used for visualizations
Number of periodic moments of breathing
Analysis
Time domain analysis
Statistical analysis
Averages
Variance
Envelope signal analysis
Counters
Frequency domain analysis
Fast Fourier Transform using a sliding window
Band pass filter
Experiment Protocol
List three times to use diaphragmatic breathing during recording session.
Schedule 1 hour recording session.
Recording of training data for breathing and movement.
Classification
Single classifier used
Combination of classifiers used for verification
Binary yes or no regarding diaphragmatic breathing
Binary yes or no as well as abdominal component of the signal to reflect attempts during rest moments

11 C. Stakeholder Interview Questions

11 C.1 Interview with Mr. Ben Bulsink

Listed below are the questions posed to Mr. Bulsink in Section 4.4.1 during the ideation phase.

What inspired you to invent the breathing tracker?

What are your perceived benefits of abdominal breathing? Were you aware of any in particular before hand?

How do you see users using the device in the short term?

How do you see users using the device in the long term?

What kind of statistics would you like to know about your own breathing?

In which scenarios would you find a breathing tracker useful?

In which scenarios would you find a breathing tracker a nuisance?

What do you think about haptic feedback being included in the device?

Are there any activities in particular you would like this to be used for or would use yourself for?

Are there any activities in particular you foresee people using this device to monitor their breathing while doing?

What kind of notification system would you find ideal for a system like this in case diaphragmatic breathing has not been performed in a while?

11 C.2 Interview with student

Listed below are the questions posed to a student of Advanced Technology at the University of Twente, who is a avid user of wearable technology in Section 4.4.2.

Have you ever used a workout trainer or other such instructional application before?

What did you like about the apps?

What did you dislike about the apps?

What is your opinion on wearable devices?

Do you currently use any wearable devices?

What do you like about wearable device? What don't you like about them?

Have you ever been aware of your breathing during an activity such as studying or walking?

Have you heard of diaphragmatic breathing?

Would you be interested in monitoring your breathing throughout the day or during activities you perform?

What kind of notification system would you find ideal for a system like this in case diaphragmatic breathing has not been performed in a while?

What kind of statistics do you like to see regarding the activity you were performing?

11 C.3 Interview with teacher

Listed below are the questions posed to a teacher of Creative Technology and researcher in Wearable Technology at the University of Twente in Section 4.4.3.

What don't you like about wearables?

What kinds of home health devices or apps have you been using? Can you mention the best and the worst if any

Can you give an example if a really well designed GUI and why you found it to be so?

What are your experiences with them?

What prompted you to start using the device?

Which parts of the Spire application did you like? Which did you dislike?

Did you encounter problems using these devices or apps?

What are you trying to achieve with these devices and apps? Answer for each one

Is there anything you do not like about the real time feedback aspect of the Spire?

During which situations would prefer real time feedback?

What would be a preferred way of feedback for you while using Ben's device?

Have you had experience with instructional feedback? What do you think about instructional feedback?

Which colours would you feel like to fit in an application?

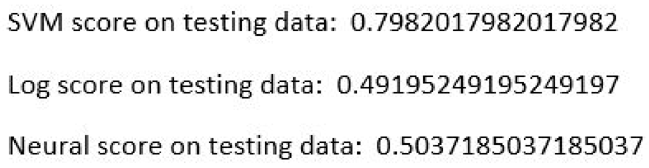
Are there any statistics in particular that you would have liked to see in the spire app that weren't there or that were there and you liked?

In what situations did you use the spire or especially pay attention to it? Did you wear it all day and just pay attention when you got notified, or did it make you subconsciously pay attention to your breathing?

11 D. Classifier Accuracy Tests

Raw RIP values

When the classifiers were only trained on the raw RIP values (prediction accuracies shown in Image 11.D.1), which included all the noise from speech and baseline drift, the SVM scored the highest (79.8%) Due to the presence of noise, and the fact that the values are not normalized, this data should not be the only data the classifiers are trained on.

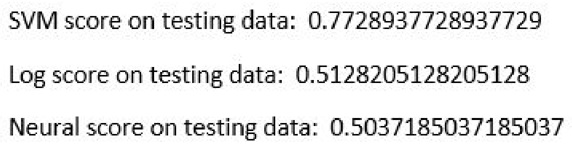


```
SVM score on testing data:  0.7982017982017982

Log score on testing data:  0.49195249195249197

Neural score on testing data:  0.5037185037185037
```

Image 11.D.1*: Breathing prediction scores of the various classifiers trained on the raw data from the bands*

Relative contributions

The contributions of the abdominal and chest band were summed and their relative contributions calculated by dividing each of them by the sum (Image 11.D.2). There is a reduction in prediction accuracy of the SVM compared to the results in Image 11.D.1 and this can be attributed to the fact that, as explained in Section 2.1.3.1.2, SVMs transform features into a higher dimensional space to construct a hyperplane to separate the classes. This means that the features can be squared or cubed, however if they are numbers less than zero, like in the case of contributions which add up to 1, then the numbers become smaller after an exponent operation is performed on them, making it in fact harder to easily separate. Therefore, relative contributions on their own will not be used to train the breathing classifiers.

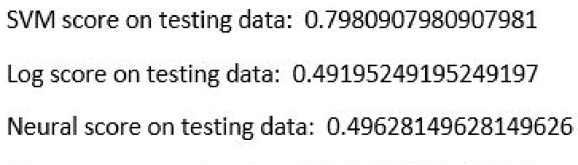


```
SVM score on testing data:  0.7728937728937729

Log score on testing data:  0.5128205128205128

Neural score on testing data:  0.5037185037185037
```

Image 11.D.2: *Breathing prediction scores of the various classifiers trained on the relative contribution of the two bands*

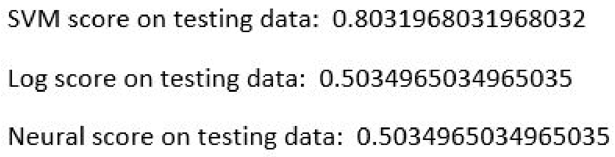
Raw data and the relative contributions

The classifiers were trained with both the raw data, and the relative contributions of the two bands alone. The results in Image 11.D.3 show that the SVM accuracy almost matches that of the raw data. It would appear that the addition of the relative contribution to the raw data does not seem to have any a noticeable effect on the accuracy of the classifiers. This could have something to do with the fact that contribution values are less than 0, and that it is difficult for classifiers to distinguish between small numbers without a large training set size. This is confirmed when the ratio of the test split is lowered to 0.1, shown in Image 11.D.4, where in all cases, the prediction accuracies increase.



```
SVM score on testing data:  0.7980907980907981
Log score on testing data:  0.49195249195249197
Neural score on testing data:  0.49628149628149626
```

Image 11.D.3: *Breathing prediction score of classifiers trained on raw data and relative contributions of the two bands with 0.9 train test split*

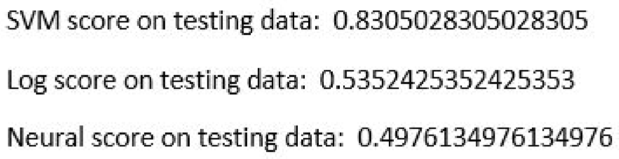


```
SVM score on testing data:  0.8031968031968032
Log score on testing data:  0.5034965034965035
Neural score on testing data:  0.5034965034965035
```

Image 11.D.4: *Breathing prediction score of classifiers trained on raw data and relative contributions of the two bands with 0.1 train test split*

Filtered and normalized data

The classifiers were trained on the filtered data (the RIP data was put through a band pass filter with range 0.2 - 0.7 Hz) from the abdominal and chest bands. The function used for the filtering also normalized the filtered data. The accuracies for the SVM has noticeably increased. This combination is worth being considered as training material for the the classifiers.

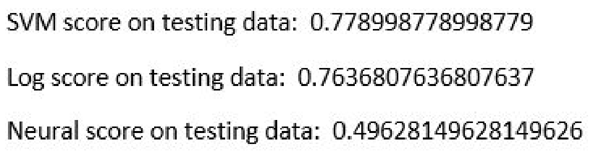


```
SVM score on testing data:  0.8305028305028305
Log score on testing data:  0.5352425352425353
Neural score on testing data:  0.4976134976134976
```

Image 11.D.5: *Breathing prediction score of classifiers trained on filtered and normalized data of each band*

Filtered data and the relative contributions

Seeing as the addition of the relative contributions to the raw data did not change the prediction accuracy very much, the relative contributions were added to the filtered data to note the difference. Comparing Image 11.D.5 with 11.D.6, it can be seen that the prediction accuracies actually reduce or stay the same for all cases except the logistic regression classifier, which has a significant increase.



```
SVM score on testing data:  0.778998778998779

Log score on testing data:  0.7636807636807637

Neural score on testing data:  0.49628149628149626
```

Image 11.D.6: *Breathing prediction scores of classifiers trained on filtered, normalized data and relative contributions of each band*

11 E. Informed Consent Form

Lead Researcher:

Name: Arnav Mundkur
Address: Hengelosestraat 248, 7521 AL, Enschede
Telephone number: +31636485224
Email address: a.mundkur@student.utwente.nl

Supervisor:

Name: Erik Faber
Address: Zilverling A112, 7500 AE, Enschede
Telephone number: +31534892041
Email address: e.j.faber@utwente.nl

Form of Consent

Purpose of the research study: The purpose of the study is to classify respiratory data to identify periods of periodic and diaphragmatic breathing. This project has two goals, to classify the data and to design a GUI for feedback presentation. The latter study is being carried out by Florian Naumilkat. The goal of the study is to train regular diaphragmatic breathing through the use of the wearable recording device, and feedback provided to the user through an interface (Florian Naumilkat's GUI project).

What you will do in the study: Participants in this study will be taught diaphragmatic breathing by a trainer during a breathing training session. Their diaphragmatic breathing will then be recorded using Respiratory Inductance Plethysmography (RIP), which involves wearing two bands; one around the abdomen and the other around the chest, that measure expansion and contraction of the area during breathing. Participants will then be instructed practice for 10 minutes at home every day and to keep an informal breathing diary where they mention if they were able to practice diaphragmatic breathing during the day.
Recording sessions will be arranged with the participants, where they will wear the bands and record data for the span of an afternoon. This is the real data collection phase and the data will be used to identify whether they used diaphragmatic breathing during the period of time the data collection was happening among other characteristics of the data that will be provided in the GUI. The data that will be collected is respiratory data using RIP as well as accelerometer data to record movement activity to provide information to help understand the respiratory data.
At the beginning of each recording session, the participant will be asked to record 10 minutes of data in total to establish a baseline, after which they may continue with their daily activities.

Time required: The study will require about 9 hours of your time. Daily practice of around 10 minutes of diaphragmatic breathing between the training session and the two recording sessions, which logistically will occur over the course of one and a half weeks or 10 days, putting total time spent on practicing breathing at around one and a half hours. During the two recording sessions that account for the majority of time spent, the participant will wear the recording bands and go about their daily routine.

Risks: Slight discomfort from wearing elastic bands around the abdomen and chest for extended periods of time. This can be alleviated by putting a piece of cloth under the bands or wearing them over a shirt. There is a small chance of temporary distraction due to the measurement bands akin to wearing a new watch for the first time. Based on the individual's experience with RIP, the decision to wear the bands over the clothes needs to be consistent, for the sake of consistency in the data.

Benefits: There are no immediate benefits for participating in this research study, however diaphragmatic breathing has many proven and noted physiological health benefits that can be experienced as a result of daily practice. The study will help understand whether the method of data collection i.e. RIP, is useful as a training device for providing an overview of breathing activity during the day.

Confidentiality: Neither the raw data (expansion and contraction of the participant's chest and abdomen, and accelerometer data), nor the processed statistics will be shared with any third parties. Features extracted such as breathing rate, peak to peak frequency and other such statistics will only be stored for the sake of providing feedback to the participant. The data will be presented anonymously in any pictures of the GUI or code.

Voluntary participation: Participation in the study is completely voluntary.

Right to withdraw from the study: A participant has the right to officially withdraw from the study at any time without penalty. The raw respiratory and accelerometer data, as well as any processed data will be deleted if it has not been anonymized and 48 hours have not passed since the recording.

How to withdraw from the study: Please contact Arnav Mundkur via the details provided below (either email, telephone or Whatsapp message). The message/phone should include the reason for the termination of the participation in the experiment.

Duration of data storage: All data including the consent forms, raw RIP and accelerometer data will be stored until the end of the project (6th July 2018 unless an extension is necessary), after which point it will be manually deleted. These files will be stored securely using password protection and be anonymized using a key mapping participant name to number that is also stored using password protection.

Payment: You will receive no payment for participating in the study.

Agreement:

I have read the information provided above thoroughly and understand what data will be collected, what it will be used for and how it will be treated if I choose to end my participation. I provide my consent for the participation in this experiment, and the collection of my personal movement (accelerometer) and respiratory (RIP) data.

Participant name: ____

Participant signature: ____ Date: ______________________

I have read the **Accidental Discovery** clause and provide my consent.

Participant name: ____

Participant signature: ____ Date: ______________________

If you have questions about the study, please feel free to contact the lead researcher or supervisor. In case you wish to contact someone unrelated to the project, details are provided below:

Name: Mrs. J.M. Strootman-Baas
Position: Secretary of the Ethical Committee of the faculty EEMCS, University of Twente
Email address: ethics-comm-ewi@utwente.nl
Telephone number: 053-489 6719

11 F. Program code

```
import matplotlib.pyplot as plt
import scipy.signal as s
from sklearn.tree import DecisionTreeClassifier
from sklearn import svm
import numpy as np
from os import path
import json

# Reads the raw data from a text file, and returns an array of elements with values:
# [R1, R2, X, Y, Z, Date-time stamp, counter] is the format of the data returned

def read_from_file(name):

    # Reads the data from the text file
    with open(name, "r") as file:
        Data = file.readlines()

    data = []   # data from the prototype stored per reading
    oneline = []    # to store all the measurements of one sample
    time = []      # to store all the times in the data
    remove = []     # to remove the unwanted labels like "Mode" and "Powering up"
    missingData = False     # used to check for missing data during the recording

    # Loop to add all elements that need to be removed to the remove list
    for i in Data:
        if i[0] == 'L':
            remove.append(i)
        elif i[0] == 'P':
            remove.append(i)
        elif i[0] == 'M':
            remove.append(i)

    # Remove the elements in the remove list from the data
    for i in remove:
        Data.remove(i)

    # Collecting the data in sets of values from R1 to Z
    for i in range(0, len(Data)):

        # Getting length of value
        length = len(Data[i]) - 1
        temp = Data[i][0:length]

        # Looking at the label of each element
        check = temp[0]

        # Checking if the label is something related to the data
        if check == "R" or check == "X" or check == "Y" or check == "Z":
            oneline.append(temp)

        # Checking if the element is a date-time stamp which is then added to the times list
        elif check == "T" or isinstance(check, str):
            # Removing the T) from the date-time stamp
            if len(temp) == 22:
                temp = temp[3:]
            elif len(temp) == 23:
                temp = temp[4:]
            time.append(temp)

        else:
            continue

        # Ending each line of rawData to be added to the data with the Z value
        for j in range(0, len(oneline)):
            # Checking if the label is Z, the L stands for live
            if "ZL)" in str(oneline[j]) or "Z)" in str(oneline[j]):
                # Removes the R1), R2), X), Y), and Z) tags from the raw data
                for k in range(0, len(oneline)):
                    if ")" in str(oneline[k]):
                        oneline[k] = oneline[k].split(")")[1]
                        oneline[k] = int(oneline[k])

                # If for any of the data entries, if R1, R2, X, Y or Z is missing, it turns the boolean to true
                if len(oneline) <= 4:
                    missingData = True

                # If the length of the data is longer than 5 for some unknown reason, that set of data is deleted
                if len(oneline) > 5:
                    oneline = []

                # Unless the set of data is empty, append it to data
                if len(oneline) != 0:
                    data.append(oneline)

                oneline = []   # Reset oneline to empty for the next sample

                # Append the latest time to the data so that samples are interrupted by a date-time stamp
                if len(time) > 0:
                    data.append(time)
                    time = []

    oldT = 0   # Value use to store the index of the previous date-time stamp
    tIndex = 0  # Index value of the current date-time stamp

    # This block removes the independent time stamps and gives each data entry a time stamp
    for i in range(0, len(data)):
        # This checks if the element in the list is a date-time stamp or not
        if len(data[i]) < 2:
            tIndex = i
            counter = 0  # Counter is used to uniquely identify each sample

            # For all data entries between the index of the old date-time stamp and the current one
            for j in range(oldT, tIndex):
                if isinstance(data[j][0], int):
                    # The time is added to each sample here
                    counter += 1
                    data[j].extend(data[i])    # data[i] is the previous time found
                    data[j].extend([counter])   # Counter to uniquely identify the sample

            oldT = tIndex   # Overriding the index of the last date-time stamp

    # Look for the last date-time stamp
    for i in range(oldT, len(data)):
        if len(data[i]) < 2:
            tIndex = i

    # Remove all the values that do not have a date-time stamp
    if tIndex == oldT:
        data = data[:tIndex]

    # Removes any empty samples or other labels missed by the previous code
    for i in data:
        if isinstance(i[0], str) or i == []:
            data.remove(i)

    return data, missingData
# ______________________________________________________________________________

# Reads the raw data from a test file and returns it along with a list of labels the same length as the data
# [R1, R2, X, Y, Z] is the format of the data returned

def train_from_file(name, label):
    with open(name, "r") as file:
        Data = file.readlines()

    data = []   # Data to be returned
    oneline = []   # To store the data of one sample
    remove = []    # List storing all the values that need to be removed

    # Loop to add all elements that need to be removed to the remove list
    for i in Data:
        if i[0] == 'L':
            remove.append(i)
        elif i[0] == 'P':
            remove.append(i)
        elif i[0] == 'M':
            remove.append(i)
    # Remove the elements in the remove list from the data
    for i in remove:
        Data.remove(i)

    for i in range(0, len(Data)):
        length = len(Data[i]) - 1
        temp = Data[i][0:length]    # Each data value of each sample

        # Checking the label of the data point to see if it is R, X, Y or Z
        check = temp[0]
        if check == "X" or check == "Y" or check == "Z" or "R" in check:
            oneline.append(temp)
        else:
            continue

        # Cutting oneline into data from one sample
        for j in range(0, len(oneline)):
            if "Z" in oneline[j]:   # End the chunk at this data point
                for k in range(0, len(oneline)):
                        if ")" in oneline[k]:
                            oneline[k] = oneline[k].split(")")[1]
                            oneline[k] = int(oneline[k])
                data.append(oneline)
                oneline = []

    # data = normalize_breathing_data_training(data)

    # Generate a list of labels of the input label
    labels = []
    for i in range(0, len(data)):
        labels.append(label)

    return data, labels
# __________________________________________________________________________________________________________

# This function extracts the R1 and R2 values for the purpose of training the breathing classifier
# [[R1, R2], [R1, R2]] is the format of the returned list

def get_breathing_data_for_training(data):

    r1vals = []    # Lists used for storing the R1 and R2 band data
    r2vals = []

    # Separating the values to filter into R1 values and R2 values
    for i in range(len(data)):
        r1vals.append(data[i][0])
        r2vals.append(data[i][1])

    # Settings for the digital bandpass filter
    samplingFrequency = 8
    lowEnd = 0.2
    highEnd = 0.7
    order = 9  # order of the filter
    nyq = 0.5 * samplingFrequency  # Nyquist frequency used to calculate the lowcut and highcut values
    lowCut = lowEnd / nyq
    highCut = highEnd / nyq

    # Calculate the coefficients for the filter function
    B, A = s.butter(order, [lowCut, highCut], output='ba', btype='band')

    # The filtered R1 and R2 values
    filteredr1 = s.filtfilt(B, A, r1vals)
    filteredr2 = s.filtfilt(B, A, r2vals)

    for i in range(len(data)):
        # Removing the accelerometer data
        for j in range(3):
            del (data[i][2])

        # Calculating the relative contributions
        sum = abs(data[i][0]) + abs(data[i][1])
        contribR1 = abs(data[i][0]) / sum  # R1 is the abdominal band
        contribR2 = abs(data[i][1]) / sum  # R2 is the chest band

        # Adding the additional breathing features to improve classification accuracy
        data[i].extend([filteredr1[i]])
        data[i].extend([filteredr2[i]])
```

```
      data[i].extend([contribR1])
      data[i].extend([contribR2])

   return data
# ____________________________________________________________________________________

# This function returns the R1 and R2 values of each sample grouped by minute for the purpose of breathing prediction
# [[[R1, R2], [R1, R2]], [[R1, R2], [R1, R2]]] is the format of the returned list

def get_breathing_data_for_classification(data):
   breathing_data = []

   for i in range(0, len(data)):   # for each minute
      oneminute = []
      for j in range(0, len(data[i])):    # for every sample in the minute
         onesample = []
         for k in range(2):
            onesample.append(data[i][j][k])
         oneminute.append(onesample)
      breathing_data.append(oneminute)

   return breathing_data
# ____________________________________________________________________________________

# This function returns the X, Y and Z values of each sample for the purpose of training the movement classifier
# [[X, Y, Z], [X, Y, Z], [X, Y, Z]] is the format of the returned list

def get_acc_data_for_training(data):
   acc_data = []

   for i in range(len(data)):
      oneline = []
      if len(data[i]) > 1:
         oneline.append(data[i][2])  # Append the X value
         oneline.append(data[i][3])  # Append the Y value
         oneline.append(data[i][4])  # Append the Z value

         # Only add it to the training data if the sample is not empty
         if len(oneline) > 0:
            acc_data.append(oneline)

   return acc_data
# ____________________________________________________________________________________

# This function returns the X, Y and Z values of samples grouped by minute for the purpose of movement prediction
# [[[X, Y, Z], [X, Y, Z], [X, Y, Z]], [[X, Y, Z], [X, Y, Z], [X, Y, Z]]] is the format of the returned list

def get_acc_data_for_classification(data):
   acc_data = []

   for i in range(len(data)):     # for every minute
      oneminute = []
      for j in range(len(data[i])):   # for every sample in every minute
         onesample = []
         onesample.append(data[i][j][2])    # Append the X value
         onesample.append(data[i][j][3])    # Append the Y value
         onesample.append(data[i][j][4])    # Append the Z value
         oneminute.append(onesample)
      acc_data.append(oneminute)

   return acc_data

# ____________________________________________________________________________________

# This function removes noise outside the 0.2-0.7 Hz frequency range for data used for training the breathing classifier
# [[R1, R2], [R1, R2]] is the format of the returned list

def remove_noise_training(breathingData):

   r1vals = []
   r2vals = []

   # Removing 0 values
   breathingData = remove_zeros_training_data(breathingData)

   # Separating the values to filter into R1 values and R2 values
   for i in range(len(breathingData)):
      r1vals.append(breathingData[i][0])
```

```
        r2vals.append(breathingData[i][1])

    # Settings for the digital bandpass filter
    samplingFrequency = 8
    lowEnd = 0.2
    highEnd = 0.7
    order = 9                  # order of the filter
    nyq = 0.5 * samplingFrequency   # Nyquist frequency used to calculate the lowcut and highcut values
    lowCut = lowEnd / nyq
    highCut = highEnd / nyq

    # Calculate the coefficients for the filter function
    B, A = s.butter(order, [lowCut, highCut], output='ba', btype='band')

    # The filtered R1 and R2 values
    filteredr1 = s.filtfilt(B, A, r1vals)
    filteredr2 = s.filtfilt(B, A, r2vals)

    # Putting the filtered R1 and R2 back into breathingData to return
    for i in range(0, len(breathingData)):
        breathingData[i][0] = filteredr1[i]
        breathingData[i][1] = filteredr2[i]

    return breathingData
# ________________________________________________________________________________________

# This function removes noise outside the 0.2-0.7 Hz frequency range for data for breathing prediction
# [[[R1, R2], [R1, R2]], [[R1, R2], [R1, R2]]] is the format of the returned list
#
def remove_noise(breathingData):

    # Lists to store the R1 and R2 values
    r1vals = []
    r2vals = []
    unfilteredData = []

    for i in range(len(breathingData)):    # for every minute
        oneline1 = []
        oneline2 = []
        forUnfiltered = []

        for j in range(len(breathingData[i])):  # for every data sample in every minute
            oneline1.append(breathingData[i][j][0])     # Append the R1 value
            oneline2.append(breathingData[i][j][1])     # Append the R2 value
            forUnfiltered.append([breathingData[i][j][0], breathingData[i][j][1]])
        unfilteredData.append(forUnfiltered)
        r1vals.append(oneline1)
        r2vals.append(oneline2)

    # Settings for the digital bandpass filter
    samplingFrequency = 8
    lowEnd = 0.2
    highEnd = 0.7
    N = 9                       # order of the filter
    nyq = 0.5 * samplingFrequency      # Nyquist frequency used to calculate the lowcut and highcut values
    lowCut = lowEnd / nyq
    highCut = highEnd / nyq

    # Calculate the coefficients for the filter function
    B, A = s.butter(N, [lowCut, highCut], output='ba', btype='bandpass')

    filtr1 = []
    filtr2 = []

    for i in range(len(r1vals)):       # For each minute

        filteredr1 = s.filtfilt(B, A, r1vals[i])    # classify all the R1 values for each minute
        filtr1.append(filteredr1)

        filteredr2 = s.filtfilt(B, A, r2vals[i])    # classify all the R2 values for each minute
        filtr2.append(filteredr2)

    # Code to test frequency filter
    # rawSums = []
    # filtfiltsums = []

    # choice = 13
    #
    # for i in range(len(r1vals)):
    #     rawSums.append(r1vals[i])
    #     filtfiltsums.append(filtr1[i])
    #
    #
    # plt.subplot(2,2,1)
    # plt.plot(rawSums[choice][200:300])
    # plt.title("Raw data")
    #
    # plt.subplot(2, 2, 2)
    # plt.plot(filtfiltsums[choice][200:300])
    # plt.title("FiltFilt")
    #
    # plt.show()

    # Putting the filtered values back into breathingData
    for i in range(len(breathingData)):       # For every minute
        for j in range(len(breathingData[i])):  # For every sample of every minute also length of every minute of data in filtr1
            breathingData[i][j][0] = filtr1[i][j]            # R1 value of the sample of the minute data
            breathingData[i][j][1] = filtr2[i][j]            # R2 value of the sample of the minute data

    return breathingData, unfilteredData

# ______________________________________________________________________________________________________

# This returns an SVM classifier trained on the training data provided as an argument

def get_trained_breathing_classifier(trainingData, labels):
    clf = svm.SVC(gamma=0.0001, C=100)     # Smaller the gamma the more accurate it is
    bd = get_breathing_data_for_training(trainingData)  # R1 and R2 data

    clf.fit(bd, labels)

    return clf

# ______________________________________________________________________________________________________

# This returns an Decision Tree classifier trained on the training data provided as an argument

def get_trained_movement_classifier(trainingData, labels):
    clf = svm.SVC(gamma=0.0001, C=100)
    td = get_acc_data_for_training(trainingData)    # X, Y, Z data for training

    clf.fit(td, labels)

    return clf

# ______________________________________________________________________________________________________

# Appends a movement label to the end of each sample calculated off the majority for each minute
# [[[R1, R2, X, Y, Z, Date-time, counter, sum, "Breathing", "Movement"]]] is the format of the returned list

def get_movement_prediction(clf, data):
    acc_data = get_acc_data_for_classification(data)

    for i in range(len(acc_data)):  # For every minute in the data
        for j in range(len(acc_data[i])):  # For every sample of every minute of data
            sample = acc_data[i][j]
            sample = np.asarray(sample)    # Converting it to an array
            sample = sample.reshape(1, -1)  # Reshaping it to reflect one sample
            decision = clf.predict(sample)
            data[i][j].extend(decision)

    for i in data:           # for every minute of data
        statCounter = 0       # Counting the samples classified as "Stationary"
        moveCounter = 0       # Counting the samples classified as "Moving"
        total = 0           # Counting the total number of samples classified
        for j in range(len(i)):    # For every sample in every minute of data
            if i[j][11] == "Moving":
                moveCounter += 1
                total += 1
            else:
                statCounter += 1
                total += 1

        movingContribution = moveCounter / total

        if movingContribution >= 0.5:   # Arbitrary threshold that can be changed
```

```
      for k in range(len(i)):
        i[k][11] = 'Moving'
    else:
      for k in range(len(i)):
        i[k][11] = 'Stationary'

  return data

# _____________________________________________________________________________________

# Appends a breathing label to the end of each sample calculated off whether the ratio of DB:CB exceeds the threshold
# [[[R1, R2, X, Y, Z, Date-time, counter, sum, "Breathing"]]] is the format of the returned list

def get_breathing_prediction(clf, data, tolerance, unfilteredData):
  breath_data = get_breathing_data_for_classification(data)

  # For the prediction, use the contributions of every entry of the data, and then append the label to the particular
  # entry.

  for i in range(len(breath_data)):  # Every minute of breathing data
    for j in range(len(breath_data[i])):  # Every data sample of each minute of breathing data
      sample = []
      sum = abs(breath_data[i][j][0]) + abs(breath_data[i][j][1])
      contribR1 = abs(breath_data[i][j][0]) / sum
      contribR2 = abs(breath_data[i][j][1]) / sum
      sample.append(unfilteredData[i][j][0])    # Raw data value
      sample.append(unfilteredData[i][j][1])
      sample.append(breath_data[i][j][0])      # Filtered data value
      sample.append(breath_data[i][j][1])
      sample.append(contribR1)              # Contribution values
      sample.append(contribR2)
      sample = np.asarray(sample)      # Converting it to an array
      sample = sample.reshape(1, -1)    # Reshape it because we are looking at one sample
      decision = clf.predict(sample)
      data[i][j].extend(decision)

  for i in data:             # For every minute of data
    dCounter = 0           # Counter for the samples classified as Diaphragmatic Breathing
    cCounter = 0           # Counter for the samples classified as Chest Breathing
    total = 0             # Total number of samples classified in each minute

    for j in range(len(i)):    # For every sample in every minute of data
      if i[j][10] == "DB":
        dCounter += 1
        total += 1
      else:
        cCounter += 1
        total += 1

    dContribution = dCounter/total  # Calculating the ratio of samples classified DB to the total samples

    if dContribution >= tolerance:  # Checks if the ratio is higher than the entered tolerance
      for k in range(len(i)):
        i[k][10] = "DB"
    else:
      for k in range(len(i)):
        i[k][10] = "CB"

  return data
# _____________________________________________________________________________________

# Returns data grouped by minute
# [[[R1, R2, X, Y, Z, Date-time stamp, counter], [R1, R2, X, Y, Z, Date-time stamp, counter]]] is the format of the
# returned list

def separate_data_by_minute(data):

  minutes = []
  minuteData = []

  # Loops through all the data to find the minutes and save them to the minutes list
  for i in data:
    for j in i:
      if isinstance(j, str):
        j, minute = str(j).split(" ")
        minute = minute[:-3]
        if minute not in minutes:
          minutes.append(minute)
    # For every minute in the list of minutes, check if the data belongs to that minute and if it does append it to a
    # list containing all the data from that minute
    for k in minutes:     # for every minute listed in the list of minutes for the data
        oneline = []
        for i in data:    # for every single sample in the data
            for j in i:
                if isinstance(j, str):
                    j, minute = j.split(" ")
                    minute = minute[:-3]
                    if minute == k:    # If the sample's minute matches the current minute in the list of minutes
                        oneline.append(i)

        minuteData.append(oneline)

    # Remove minute data that is too short for the filtfilt function in remove_noise()
    for i in minuteData:
        if len(i) < 100:
            minuteData.remove(i)

    return minuteData

# ____________________________________________________________________________________________

# Calculates the sum of R1 and R2 for each sample, and appends it to each sample
# [R1, R2, X, Y, Z, Date-time stamp, counter, sum]

def get_breathing_sums(data):
    sums = []   # List of sums

    # Calculate the sum of R1 and R2 for every sample in every minute in the data
    for minute in data:
        oneline = []
        for j in minute:
            sum = j[0] + j[1]
            oneline.append(sum)
        sums.append(oneline)

    # Appending the sum to each sample in each minute
    for i in range(len(data)):  # for every minute
        for j in range(len(data[i])):
            data[i][j].append(sums[i][j])

    return data

# ____________________________________________________________________________________________

# Returns the breathing frequency of every minute of data appended to each sample of data
# [R1, R2, X, Y, Z, Date-time stamp, counter, sum, breathingFrequency]

def breathing_frequency(data):
    sums = []

    # Calculating the sum of R1 and R2 for every sample in every minute
    for minute in data:
        oneline = []
        for j in minute:
            sum = j[0] + j[1]
            oneline.append(sum)
        sums.append(oneline)

    freqs = []     # Stores the frequencies measured in the periodogram
    powers = []    # Stores the powers of the frequencies measured in the periodogram

    for i in sums:     # every minute
        f, p = s.periodogram(i, fs=8.0)
        freqs.append(f)
        powers.append(p)

    maxfreqs = []

    # Look for the highest power in the powers list, find its index and find the corresponding frequency
    for f in range(len(powers)):
        maxf = np.max(powers[f])
        index = list(powers[f]).index(maxf)
        maxfreqs.append(freqs[f][index] * 60)  # Multiplication by 60 to make it breaths per minute

        # Code to plot the periodogram powers and frequency
    # plt.plot(freqs[f], powers[f])
    # plt.xlabel("Frequency (Hz)")
    # plt.ylabel("RMS amplitude (V)")
    # plt.show()

  # Append the frequency of the entire minute to each sample of the minute for every minute
  for i in range(len(data)):  # Per minute
    for j in range(len(data[i])):    # Every entry in the minute
      freq = maxfreqs[i]
      data[i][j].extend([freq])

  return data

# ________________________________________________________________________________________

# Appends the label "Repetitive" or "Non-Repetitive" depending on the variance of the breathing frequency of the minute
# compared to the previous two minutes
# [R1, R2, X, Y, Z, Date-time stamp, counter, sum, breathingFrequency, "Repetitive"]

def repetitive_labeler(data, buffersize):

  # Buffersize is the argument passed in when the program is launched
  for j in range(buffersize, len(data)):
    buffer = []    # List to store buffer data

    # Find the standard deviation of the previous two minutes of data
    for k in range(buffersize):
      for l in range(len(data[j-buffersize-k])):
        buffer.append(data[j-buffersize-k][0][8])   # This is the breathing frequency

    bufferAverage = np.average(buffer)     # Calculating the average of the two minutes of data
    bufferStd = np.std(buffer)           # Calculating the standard deviation of the two minutes of data

    current = data[j][0][8]             # The current minute's breathing frequency

    if (bufferAverage - bufferStd) <= current <= (bufferAverage + bufferStd):
      for i in data[j]:
        i.append("Repetitive")
    else:
      for i in data[j]:
        i.append("Non-Repetitive")

  # Label the data in the buffer as non-repetitive
  for i in range(buffersize):
    for j in data[i]:
      j.append("Non-Repetitive")

  return data

# ________________________________________________________________________________________

# Returns an array with information regarding the breathing in the following format:
# [[(Starting and Ending time), (Quality factor), (Total time spent breathing diaphragmatically),
# (Number of diaphragmatic periods), (Time of the longest diaphragmatic moment),
# (Duration of longest diaphragmatic moment)][(Time of each diaphragmatic moment), (Duration of diaphragmatic moment),
# (Abdominal contribution)], [[(time, breathingFrequency)], [(time, breathingFrequency)]]

def db_features(data):

  startTimes = []    # List to store start times of all the diaphragmatic moments
  stopTimes = []     # List to store end times of all the diaphragmatic moments
  inMoment = False    # Boolean to check whether there is currently a diaphragmatic moment going on
  DBPeriods = []     # List storing details of diaphragmatic periods
  answer = []       # List storing the features

  for i in range(len(data)):

    # Check if the sample is labeled Diaphragmatic breathing and that a period isn't currently under way
    # and sets inMoment to true if it isn't true, to signal the beginning of a diaphragmatic moment
    if data[i][0][10] == 'DB' and not inMoment:
      startTimes.append(data[i][0][5])    # Add the start time of the diaphragmatic moment to the start times list
      inMoment = True

    # If the sample is labeled Chest breathing, then the moment has been ended and the stop time is added
    # to the list of stop times, and the inMoment boolean is set to false
    if data[i][0][10] == 'CB' and inMoment:
      stopTimes.append(data[i][0][5])
      inMoment = False
```

```
# Adds the last time of the recording if there was DBM when the recording ended
if len(startTimes) > len(stopTimes):
    stopTimes.append(data[len(data)-1][len(data[len(data)-1])-1][5])

# No signifies the unwanted split part of the string
for i in range(len(startTimes)):
    beginning = startTimes[i]
    no, sd = stopTimes[i].split(" ")
    no, ft = beginning.split(" ")
    startTimes[i] = ft[:-3]     # replacing the start date and time with just the time

    # Calculating the minutes between the start and the stop
    firsthour = int(ft[:2])
    secondhour = int(sd[:2])
    firstMinute = int(ft[3:5])
    secondMinute = int(sd[3:5])

    # Math regarding calculation of time in minutes between two time stamps
    first = (firsthour * 60) + firstMinute
    second = (secondhour * 60) + secondMinute
    duration = second - first
    if duration < 0:
        duration = duration + 1440
    entry = str(beginning)[:-3] + "-" + str(sd)[:-3]
    # Append the start and end time as well as the duration to the DB periods list.
    DBPeriods.append([entry, duration])

# Replacing all the date-time stamps in stop times with just times
for i in range(len(stopTimes)):
    choice = str(stopTimes[i]).split(" ")
    if len(choice) > 1:
        chosen = choice[1][:-3]
        stopTimes[i] = chosen

# Find any periods that have a length of less than 2 minutes
lessThanTwo = []

for i in range(len(DBPeriods)):
    if DBPeriods[i][1] < 3:
        lessThanTwo.append(DBPeriods[i])

for i in range(len(lessThanTwo)):
    DBPeriods.remove(lessThanTwo[i])

# Find the index of the longest moment and total time spent breathing diaphragmatically
totalDB = 0
longestIndex = 0

for i in range(len(DBPeriods)):
    totalDB += DBPeriods[i][1]
    if DBPeriods[i][1] > DBPeriods[longestIndex][1]:
        longestIndex = i

# Calculate the quality factor for diaphragmatic breathing
quality_factor = totalDB / total_time_elapsed(data)

# Calculating the average diaphragmatic breathing period duration
lengths = []
for i in DBPeriods:
    lengths.append(i[1])

averageDuration = 0

if len(lengths) > 0:
    averageDuration = np.average(lengths)

DBmoments = len(DBPeriods)

# Appending the Number of diaphragmatic moments, the times of the longest moment and the longest moment.
oneline = []
oneline.append(("Start and end:", get_total_recording_time(data)))
oneline.append(("Session quality factor:", quality_factor))
oneline.append(("Total time spent DB in minutes:", totalDB))
oneline.append(("Number of DB periods:", DBmoments))

# Making sure that there are moments to append
if len(startTimes) > 0:
    oneline.append(("Times of longest DB moment:", DBPeriods[longestIndex][0]))
    oneline.append(("Duration of longest DB moment:", DBPeriods[longestIndex][1]))
    oneline.append(("Average DB moment duration:", averageDuration))
  answer.append(oneline)

  # Calculating the average of the abdominal contribution of every sample in each diaphragmatic breathing period

  for k in range(len(DBPeriods)):
    sum = 0
    counter = 0
    for i in range(len(data)):        # for every minute in the data
      for j in range(len(data[i])):   # for every sample in every minute of the data
        time = str(data[i][j][5]).split(" ")
        if len(time) > 1:
          choice = time[1]
          hour = int(choice[:2])
          minute = int(choice[3:5])

          # Getting the hour and minute of the start of the DB period
          downhour = int(startTimes[k][:-3])
          downminute = int(startTimes[k][-2:])

          # Getting the hour and minute of the end of the DB period
          uphour = int(stopTimes[k][:-3])
          upminute = int(stopTimes[k][-2:])

          # Check if the current sample is in a minute in the range of the beginning and the end of the
          # diaphragmatic breathing period
          if downhour <= hour <= uphour and downminute <= minute <= upminute or 0 <= minute <= upminute:
            if data[i][j][10] == "DB":
              contribution = abs(data[i][j][0]) / (abs(data[i][j][0]) + abs(data[i][j][1]))
              sum += contribution
              counter += 1

    # Only calculate the average if the counter actually counts minutes to avoid divide by 0 exception
    if counter > 0:
      average = sum/counter
      DBPeriods[k].append(average)

  answer.append(DBPeriods)

  return answer

# ______________________________________________________________________________________________________________

# Returns the time period of the entire recording in minutes

def total_time_elapsed(data):

  starthour = int(data[0][0][5][11:13])
  startminute = int(data[0][0][5][14:16])
  endhour = int(data[len(data) - 1][len(data[len(data) - 1]) - 1][5][11:13])
  endminute = int(data[len(data) - 1][len(data[len(data) - 1]) - 1][5][14:16])
  start = (starthour * 60) + startminute
  end = (endhour * 60) + endminute
  totalTime = end - start

  if totalTime < 0:
    totalTime = totalTime + 1440

  return totalTime

# ______________________________________________________________________________________________________________

# Write the features to a txt file for the interface

def write_features_to_file(features, name):

  filename = "output_to_Flo/" + str(name)+".txt"

  # Create a file if it doesn't exist with the inputted name
  file = open(filename, "w+")

  #Writing to the file
  for i in range(len(features)):
    if i == 1:
      file.write("[Times of the diaphragmatic breathing, Duration of the diaphragmatic breathing period,"
           " Average abdominal contribution during period]")
      file.write("\n")
      file.write(str(features[i]))
      file.write("\n")
    else:
      file.write(str(features[i]))
      file.write("\n")
  file.close()

  # Code to export data as JSON file
  # filename = "output_to_Flo/" + str(name) + ".json"
  # data = {}        # Statistics, Diaphragmatic periods, Breathing frequencies
  # data["Statistics"] = []
  # data["Diaphragmatic periods"] = []
  # data["Breathing frequencies"] = []
  #
  # # Dealing with the statistics
  # for i in range(len(features[0])):
  #    firstHalf = features[0][i][0][:-1]
  #    secondHalf = features[0][i][1]
  #    data["Statistics"].append({firstHalf: secondHalf})
  #
  #
  # for i in range(len(features[1])):
  #    first = features[1][i][0]
  #    second = features[1][i][1]
  #    third = features[1][i][2]
  #    data["Diaphragmatic periods"].append({"Times of DB moment": first, "Duration of DB moment": second,
  #                                   "Contribution from abdominal region": third})
  #
  # for i in range(len(features[2])):
  #    first = features[2][i][0]
  #    second = features[2][i][1]
  #    data["Breathing frequencies"].append({"Minute": first, "Breathing frequency": second})
  #
  #
  # with open(filename, "w+") as outfile:
  #    json.dump(data, outfile)

# ______________________________________________________________________________________________

# Removes zeros from the training data. This function is called recursively until there are no more zeros in the data

def remove_zeros_training_data(data):

  # Split the data before and after the encountered zero and put them together
  for i in range(len(data)):
    if data[i][0] == 0 or data[i][1] == 0:
      first = data[:i]
      second = data[(i+1):]
      first.append(second)
      remove_zeros_training_data(first)

  return data

# ______________________________________________________________________________________________

# Returns the range of times for the entire data set as a string for the feature description

def get_total_recording_time(data):
  startRecording = data[0][0][5]
  endRecording = data[len(data) - 1][len(data[len(data) - 1]) - 1][5][11:]
  recordingTime = startRecording + "-" + endRecording

  return recordingTime
# ______________________________________________________________________________________________

# Returns the a list of lists consisting of every minute and the average breathing frequency in that minute

def get_breathing_frequency_features(data):

  # ij[8] is the breathing frequency per sample
  timesAndFrequencies = []

  # For every minute of data look at the time and breathing frequency of the first sample
  for i in data:
    time = i[0][5]
    frequency = i[0][8]
    timesAndFrequencies.append([time, frequency])
  return timesAndFrequencies

# ____________________________________________________________________________________________________

# This function compiles test data from files and returns training data and labels for breathing and movement

def get_training_data():

  # Load stationary data and get a list of labels equally long
  stationaryData, stationaryDataLabels = train_from_file("data/stationaryData.txt", "Stationary")
  # stationaryData, stationaryDataLabels = train_from_file("training_data/Flo/stationary.txt", "Stationary")

  # Load movement data and get a list of labels equally long
  # movementData, movementDataLabels = train_from_file("training_data/Flo/moving.txt", "Moving")
  movementData, movementDataLabels = train_from_file("data/movementData.txt", "Moving")

  # Compiling the data for motion and the labels for motion into one list
  for i in movementData:
    stationaryData.append(i)

  for j in movementDataLabels:
    stationaryDataLabels.append(j)

  # Load diaphragmatic breathing data and get a list of labels equally long
  diaphragmaticBreathingData, diaphragmaticBreathingDataLabels = train_from_file("data/DB.txt", "DB")
  # diaphragmaticBreathingData, diaphragmaticBreathingDataLabels = train_from_file("training_data/Flo/DB.txt", "DB")

  # Load chest breathing data and get a list of labels equally long
  # chestBreathingData, chestBreathingDataLabels = train_from_file("training_data/Flo/CB.txt", "CB")
  chestBreathingData, chestBreathingDataLabels = train_from_file("data/CB.txt", "CB")

  # Compiling the data for motion and the labels for motion into one list
  for i in chestBreathingData:
    diaphragmaticBreathingData.append(i)

  for j in chestBreathingDataLabels:
    diaphragmaticBreathingDataLabels.append(j)

  return diaphragmaticBreathingData, diaphragmaticBreathingDataLabels, stationaryData, stationaryDataLabels

# ____________________________________________________________________________________________________

# Returns the parameters for running the program such as which file to read the session data from, which file to
# save the results to, how many minutes long the buffer should be for the repetitive breathing calculation and
# how strict the breathing classification should be

def get_info():
  tolerance = 0.0
  bufferSize = 0
  correctTolerance = False
  correctBuffer = False
  correctFileName = False

  # Ensures that the path entered exists and that no attempt was made to go up in the directory
  while not correctFileName:
    fileName = str(input("Enter the name of the raw data file to be read with: data/____.txt \n"))
    check = fileName[0]
    if check == 'd' and path.exists(fileName):
      correctFileName = True
    else:
      correctFileName = False
      print("File does not exist! \n")

  saveTo = input("Enter the title of the text file you want to save the features to:  \n")

  # Ensures that the entered buffer isn't smaller than or 0 and not greater than 7 minutes
  while not correctBuffer:
    bufferSize = input("Enter the number (whole number) of minutes of buffer for periodic breathing "
                       "calculation between 0 and 7 minutes: \n")

    if str(bufferSize).isdigit():
      if int(bufferSize) <= 0 or int(bufferSize) > 7:
        correctBuffer = False
        print("Number entered wasn't within the required range! \n")
      else:
        correctBuffer = True
    else:
            correctBuffer = False
            print("Please enter a number and not a string! \n")

    # Ensures the entered tolerance is between 0.0 and 1.00
    while not correctTolerance:
        tolerance = input("Enter the tolerance for the breathing and movement classification between 0.50 and 0.80: \n")

        try:
            t = float(tolerance)
            if t < 0.50 or t > 0.80:
                correctTolerance = False
                print("Number entered wasn't within the required range! \n")
            else:
                correctTolerance = True
        except ValueError:
            print("Please enter a float and not a string! \n")



    print("Processing.....")

    rawData, missingData = read_from_file(fileName)

    return rawData, missingData, int(bufferSize), float(tolerance), saveTo

# ____________________________________________________________________________________________________

# Small function to call the functions responsible for feature extraction and to append their answers to save to file

def extract_features(data):
    momentsAndTimes = db_features(data)
    timesAndFrequencies = get_breathing_frequency_features(data)
    momentsAndTimes.append(timesAndFrequencies)

    return momentsAndTimes

# ____________________________________________________________________________________________________

# Main funtion that runs the data processing

def run(missingData, rawData, svm, tree, bufferSize, tolerance, saveTo):

    if not missingData:
        separatedData = separate_data_by_minute(rawData)
        filteredData, unFilteredData = remove_noise(separatedData)

        breathsums = get_breathing_sums(filteredData)
        bf = breathing_frequency(breathsums)
        repetitive = repetitive_labeler(bf, bufferSize)

        bPredicts = get_breathing_prediction(svm, repetitive, tolerance, unFilteredData)
        mPredicts = get_movement_prediction(tree, bPredicts)

        breathing_features = extract_features(mPredicts)
        write_features_to_file(breathing_features, str(saveTo))

    else:
        print("Sorry there is some data missing from the R1, R2, X, Y or Z band")

# ____________________________________________________________________________________________________

# This function normalizes all the data from a breathing session for classification. Returns normalized breathing
# values for R1 and R2 values that are unfiltered

def normalize_breathing_data(data):

    for i in data:  # every minute
        r1data = []
        r2data = []
        for j in range(len(i)):
            r1data.append(i[j][0])
            r2data.append(i[j][1])

        meanr1 = np.mean(r1data)
        meanr2 = np.mean(r2data)
        stdr1 = np.std(r1data)
        stdr2 = np.std(r2data)
    # Normalization
    for k in range(len(i)):
      i[k][0] = ((i[k][0] - meanr1)/stdr1)
      i[k][1] = ((i[k][1] - meanr2)/stdr2)

  return data

# ____________________________________________________________________________________________

# This function standardizes all the raw data for training

def normalize_breathing_data_training(data):
  r1data = []    # Storing R1 values
  r2data = []    # Storing R2 values

  for i in data:  # Every sample
    r1data.append(i[0])
    r2data.append(i[1])

  # Calculating the means and standard deviations of the R1 and R2 values
  meanr1 = np.mean(r1data)
  meanr2 = np.mean(r2data)
  stdr1 = np.std(r1data)
  stdr2 = np.std(r2data)

  # Normalization
  for i in data:
    if i[0] != 0.0 and i[1] != 0.0:
      i[0] = ((i[0] - meanr1)/stdr1)
      i[1] = ((i[1] - meanr2)/stdr2)

  return data
```