Measuring the Signal Quality of Respiratory Effort Sensors for Sleep Apnea Monitoring

A Metric Based Approach

Fredrik Løberg

Thesis submitted for the degree of
Master in Informatics: Programming and Networks
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2018

Measuring the Signal Quality of Respiratory Effort Sensors for Sleep Apnea Monitoring

A Metric Based Approach

Fredrik Løberg

Measuring the Signal Quality of Respiratory Effort Sensors for Sleep Apnea Monitoring

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

Sleep apnea is a very common, yet severely under-diagnosed disorder characterized by reoccurring periods of shallow or paused breathing during sleep. If a breathing disruption causes the oxygen saturation in the blood to become too low, the brain will force an awakening to resume normal breathing. These awakenings are often very brief, making it unlikely for the sufferer to remember continuously waking up at night. The gold standard for diagnosing sleep apnea is *polysomnography*, which is a sleep study requiring the subject to spend the night in a laboratory with many physiological sensors attached to the body. This process is very resource demanding, and also very tedious and uncomfortable for the patients. Instead of providing alternatives to traditional polysomnography, our objective is to allow people to perform the first step towards a sleep apnea diagnosis at home. The core idea is to drastically reduce the cost and number of required sensors by utilizing smartphones, low-cost consumer grade sensors, and data mining techniques. Nevertheless, the realization of this idea assumes that the low-cost consumer grade sensors produce signals of adequate quality.

In this thesis, we evaluate the signal quality of four respiratory effort sensors: a *piezoelectric effort belt* (PZT) from BITalino, an *impedance plethysmography* (IP) sensor from Shimmer, a *respiratory inductance plethysmography* (RIP) sensor (RespiBAN) from biosignalsplux, and a strain-gauge sensor (FLOW) from SweetZpot. We use a RIP sensor from NOX Medical as the gold standard. Instead of recreating the setting of traditional polysomnography, we design a sixteen-minute signal capture procedure to simulate epochs of disrupted breathing, which can be performed during wakefulness. With this procedure, we capture data from a total of twelve (BITalino and Shimmer) and eleven (RespiBAN and FLOW) external subjects, resulting in a total of 212 different signals for quality evaluation. Our signal quality evaluation approach is based on the breath detection accuracy metrics *sensitivity*, *positive predictive value* (PPV), and *clean minute proportion* (CMP), along with the breath amplitude accuracy metric *weighted absolute percentage error* (WAPE). These metrics are closely related to how apneic and hypopneic episodes are scored by medical personnel, making it trivial to reason about their interpretation.

Our results show that false breaths are the primary concern affecting the breath detection accuracy of BITalino, Shimmer, and RespiBAN. Respectively, the sensitivity of BITalino, Shimmer, RespiBAN, and FLOW is 99.61%, 98.53%, 98.41%, and 98.91%. Their PPV is 96.28%, 96.58%, 90.81%, and 98.81%. Their CMP is 60.93%, 71.72%, 49.50%, and 73.08%. Finally, their WAPE is 13.82%, 16.89%, 13.60%, and 8.75%. The supine (back) position is consistently showing the overall best signal quality, and while both BITalino and Shimmer show a correlation between signal quality and *body mass index* (BMI), the supine position is less affected overall compared to the side position.

Acknowledgments

Above all, I would like to thank my supervisors, Professor Dr. Vera Goebel and Professor Dr. Thomas Plagemann, for their excellent guidance throughout the work of this thesis. Their commitment and encouragement are highly appreciated. I would also like to direct my gratitude towards all the subjects who volunteered to participate in the experiments. Without them, the work of this thesis would surely have come inadequately short.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Background and Motivation

Sleep apnea is a disorder characterized by reoccurring periods of shallow or paused breathing during sleep. If a breathing disruption causes the oxygen content (oxygen saturation) in the blood to become too low, the brain will force an awakening to resume normal breathing. These awakenings are often very brief, making it unlikely for the sufferer to remember continuously waking up at night. Repeated awakenings inhibit deep sleep, resulting in daytime sleepiness and fatigue. If untreated, sleep apnea can lead to serious health implications for the individual, and, in the worst case, even death if the person is unable to wake up. Sleep apnea is linked to serious diseases such as diabetes, hypertension (high blood pressure), heart disease, stroke, depression, and anxiety (Young et al. 2004), (Punjabi 2008), and (Huang et al. 2008).

Sleep apnea is a very common, yet severely under-diagnosed sleep disorder. It is estimated that around 25% of all middle-aged Norwegians are at high risk of having obstructive sleep apnea (Hrubos-Strøm et al. 2011), yet approximately 70–80% of all cases are expected to be undiagnosed (Punjabi 2008). Studies show that sleep apnea sufferers are about two to three times more likely to be involved in traffic accidents because of the severe sleep deprivation (McNicholas 2013). Without a recollection of the nightly awakenings, and the primary symptom being daytime sleepiness, the disorder may easily remain unnoticed. Feeling tired can be normal for many people for various reasons, and maybe even more so for long-time sufferers of sleep apnea. A study by Van Dongen et al. (2003) suggests that people, in general, are mostly unaware of being sleep deprived, which further substantiates the claim that people often are unaware of having the disorder.

The gold standard and traditional approach of diagnosing sleep apnea is with the use of *polysomnography*. Polysomnography is a sleep study which requires the subject to spend the night in a sleep laboratory with a wide range of physiological sensors attached to the body. This includes sensors for electroencephalography (EEG), electrocardiography (ECG), electromyography (EMG), electrooculography (EOG), respiratory effort from the chest (thorax) and abdomen, nasal airflow, and oxygen saturation (SpO_2) (Tripathi 2008). A sleep technologist is required to manually monitor the procedure and evaluate the results. Sleeping in an artificial and unfamiliar environment with this many sensors attached to the body can for many people feel very uncomfortable. As a result, the threshold for a potential patient to seek a diagnosis is high. In addition, this kind of sleep study is very resource demanding as it requires both expensive equipment, a suited laboratory, and trained medical personnel to manually monitor and analyze the results; making it impossible to prescribe polysomnography for everybody at risk of having sleep apnea. Portable monitoring devices have been developed to enable sleep monitoring at home without the guidance of medical personnel. However, the number of sensors attached to the body is usually not reduced too much, and the recorded signals still need to be manually evaluated by an expert before an eventual diagnosis can be determined. Additionally, these devices are usually priced way above what an average person can afford to pay on their own.

Instead of providing alternatives to traditional polysomnography, our objective is to allow people to perform the first step towards a sleep apnea diagnosis at home. The core idea is to drastically reduce the cost and number of required sensors by utilizing smartphones, low-cost consumer grade sensors (e.g., from BITalino or Shimmer), and data mining techniques. A potential sleep apnea sufferer should be able to buy an arbitrary sensor of their own choice and use that along with a smartphone to test for sleep apnea on their own. The recorded data should be analyzed by data mining techniques to automatically detect apnea events and then potentially recommend that the person should visit a physician. Furthermore, a physician should be able to use the recorded data as a foundation to better decide whether polysomnography should be performed or not.

For this to be realistic, a few requirements need to be fulfilled. Firstly, the equipment needs to be affordable and easy to use. Secondly, the user should not be bound to any specific equipment but rather be able to choose the specifics (e.g., what kind of sensor/smartphone, etc.) on their own. Thirdly and most importantly, the produced signals must be of adequate quality. Kristiansen et al. (2018) show that the quality of the signals has a huge impact on the performance of the data mining classifiers for apnea detection. In this study, they use physiological data from two databases of different quality, namely the Apnea-ECG and MIT-BIH databases from PhysioNet. The accuracy of the data mining classifiers for all signal combinations is in the range 90.6%–96.6% for the Apnea-ECG database and 58.2%–73.1% for the MIT-BIH database. Clearly illustrating the importance of data quality.

1.2 Problem Statement

In contrast to certified medical grade equipment, consumer grade electronics are usually significantly cheaper and lack formal testing. The resulting assumption is that the signal produced by cheaper sensors are also analogously of lower quality as well. Even if we accept this assumption, it does not mean that these kinds of sensors cannot be used at all during an initial test for sleep apnea. Therefore, the overall problem statement tumbles down to one enclosing question:

- Are cheaper consumer grade sensors *good enough* for an *initial* sleep apnea test?

The scope of this question is, however, too broad to fit in one thesis alone, so we have to break it down further. There are many different types of sensors used for sleep apnea monitoring, and each type captures very different kinds of data. As a result, any methods used to assess the signal quality will also be very different. Therefore, we limit the focus in this work to one kind of sensor, namely *respiratory effort sensors*. Respiratory effort sensors monitor the movement (or effort) associated with breathing, and are often belts strapped around the thorax and abdomen of the subject atop of clothing. This makes these kinds of sensors very easy to use without the help of medical personnel. Thus, they are a good candidate for our main objective. In addition, Kristiansen et al. (2018) show that respiratory effort from either the thorax or abdomen alone provides a very good classifier performance, with an accuracy of 92.9% and 72% for the abdominal signals from the Apnea-ECG and MIT-BIH databases, respectively.

There have been conducted a few studies which assess the signal quality of different types of respiratory effort sensors, e.g., (Vaughn and Clemmons 2012), (Cantineau et al. 1992), (Brouillette et al. 1987), (Whyte et al. 1991), (Adams et al. 1993), and (Cohn et al. 1982). However, the focus in these studies lies primarily on measuring the signal quality of a given type of technology, and not specifically on the signal quality of cheaper consumer grade equipment. There have been conducted studies in that regard for other types of sensors, e.g., ECG sensors (Silva et al. 2015), but to the best of our knowledge not specifically for respiratory effort sensors.

The gold standard sensor for measuring airflow is a *pneumotachograph*, which is usually a mask placed over the mouth and nose (Berry et al. 2012). In related work, any quality evaluation of respiratory effort sensors is most often conducted with a pneumotachograph as the gold standard. Unfortunately, one of our limitations is that we do not have access to a pneumotachograph, and, therefore, have to measure the signal quality using other means.

The overall problem statement breaks down to how we can measure the signal quality of respiratory effort sensors, which we address in the following questions:

1. Which metrics are appropriate?
2. In which setting should we capture the signal data?
3. How can we measure the signal quality with our limited set of resources?
4. How good are the BITalino sensors?
5. How good are the (medical grade) Shimmer sensors?
6. How good are the (medical grade) RespiBAN sensors?
7. How good are the FLOW sensors?

1.3 Approach

With respiratory effort sensors from BITalino, Shimmer, biosignalsplux (RespiBAN), SweetZpot (FLOW), and NOX Medical at our disposal, we approach the problem statement in three parts:

- Determine how the signal quality of respiratory effort sensors can be measured in relation to sleep apnea.
- Design and execute an experiment involving external subjects to measure the signal quality of the target sensors.
- Evaluate the results of the experiment and signal quality of the target sensors.

The first part focuses on what a good signal quality represents and how we can measure the quality of a respiratory effort signal (i.e., which metrics to use). It is important to emphasize that the signal quality should be measured in relation to sleep apnea monitoring. Respiratory effort sensors are used in a variety of other contexts as well, and measuring an aspect of quality which is not linked to a sensor's performance of sleep apnea monitoring potentially yields a misleading result.

We have a total of four different types of respiratory effort sensors at our disposal: *respiratory inductive plethysmography* (RIP), *piezoelectric belts* (PZT), strain-gauge belts, and *impedance plethysmography* (IP). The former three are belts strapped around the thorax and abdomen, while the latter uses electrodes attached to the skin around the thorax. Of these, only the RIP type is recommended by the *American Academy of Sleep Medicine* (AASM) for sleep apnea monitoring (Berry et al. 2012). For our experiments, we use a *BITalino Plugged Kit BLE* (BITalino 2018e) with PZT type belts, a *Shimmer ECG unit* (Shimmer 2018b) with an IP type sensor, a *biosignalsplux RespiBAN* (biosignalsplux 2018b) with a RIP type belt, a *SweetZpot FLOW* (SweetZpot 2018) with a strain-gauge belt, and a *NOX T3* (NOX Medical 2018a) with RIP type belts. These are hereafter referred to as *BITalino*, *Shimmer*, *RespiBAN*, *FLOW*, and *NOX*, respectively. We measure the signal quality of the BITalino, Shimmer, RespiBAN, and FLOW sensors by comparing them to our gold standard signal, NOX (i.e., our alternative to a pneumotachograph).

In the second part, we design an experiment to record data with these sensors from various external subjects. This design involves how long each signal capture should be, what positions the subject should undertake, and what actions the subject may perform throughout the procedure. The goal is, in other words, to gather as many representative signal captures as we need to make the results as reliable as possible. We derive the number of required subjects by studying related work.

In the last part, we evaluate the results of the experiment and signal quality of the sensors by analyzing the recorded signals. In addition to a summarized signal quality evaluation of the sensors, we also evaluate if there exist any trends in the data as well, such as the effect of *body mass index* (BMI).

1.4 Scope

The scope of this thesis covers how the signal quality of respiratory effort sensors can be measured in relation to sleep apnea, and a signal quality evaluation of the BITalino, Shimmer, RespiBAN, and FLOW sensors. Due to our limited time frame, we do not analyze any correlation between the signal quality obtained using the metrics we describe to the accuracy of the data mining techniques used by Kristiansen et al. (2018). Such an analysis would require capturing data from real sleep apnea sufferers and medical personnel to annotate and score all episodes of disrupted sleep. Thus, this is instead a part of future work.

Please note that we did not receive the RespiBAN and FLOW sensors until the very end of this work. This means that the thesis is mostly concerned with the BITalino and Shimmer sensors until the last part of Chapter 7. Additionally, the signals from the RespiBAN and FLOW sensors are not collected from all the same subjects as the BITalino and Shimmer signals.

1.5 Outline

This master's thesis is structured as follows:

- **Chapter 2 — Sleep Apnea**
  In this chapter, we present an overview of sleep apnea in general, with emphasis on the physiological signals used to diagnose the disorder.

- **Chapter 3 — Measuring Data Quality**
  This chapter gives an overview of how the quality of data can be measured, or more precisely, quantified. The emphasis lies primarily on signals from respiratory effort sensors, but the methods can be applied to other types of data as well.

- **Chapter 4 — Requirement Analysis**
  This chapter presents a requirement analysis which includes what a good signal quality represents in relation to sleep apnea, and different requirements for the experiment design (e.g., number of subjects). Furthermore, we also give an overview of the various sensor platforms we evaluate.

- **Chapter 5 — Design**
  This chapter presents how we measure the quality of the sensors (i.e., metrics), and the design of the experiment. The experiment design is based on preliminary testing and includes how long each signal capture should be, what positions the subject should undertake, and what actions the subject may perform throughout the procedure.

- **Chapter 6 — Implementation**
  In this chapter, we present our Python implementation of the metrics and the signal preprocessing steps.

- **Chapter 7 — Evaluation**
  In this chapter, we evaluate the result of the experiment and quality of the sensors by analyzing the recorded signals. We also give a review of the metrics and a comparison with related work.

- **Chapter 8 — Conclusion**
  Finally, we conclude our findings, provide a critical assessment of the work and methods, and discuss potential future work.

Chapter 2

Sleep Apnea

In this chapter, we present an overview of sleep apnea in general, with emphasis on the physiological signals used to diagnose the disorder. We begin by describing the characteristics of sleep apnea in Section 2.1, followed by a description of its different variations in Section 2.2. The most common symptoms associated with the disorder are given in Section 2.3. Next, how sleep apnea is diagnosed today, along with the various types of physiological signals used during a diagnostic procedure are presented in Section 2.4. Finally, we conclude the chapter in Section 2.5.

2.1 Characteristics

Sleep apnea is a disorder characterized by reoccurring periods of either shallow or paused breathing during sleep. If a breathing disruption causes the oxygen content (oxygen saturation) in the blood to become too low, the brain will force an awakening to resume normal breathing. These awakenings are often very brief, making it unlikely for the sufferer to remember continuously waking up at night. Repeated awakenings inhibit deep sleep, resulting in daytime sleepiness and fatigue. Untreated sleep apnea can lead to serious complications. Most notable is that it leads to daytime sleepiness, which in turn results in a lower quality of life. Sleep apnea can also in more severe cases lead to diabetes, hypertension (high blood pressure), heart disease, stroke, depression, and anxiety (Young et al. 2004), (Punjabi 2008), and (Huang et al. 2008). In other words, a proper sleep pattern is crucial for both physical as well as mental health.

Sleep apnea is a very common, yet severely under-diagnosed disorder. It is estimated that around 25% of all middle-aged Norwegians are at high risk of having obstructive sleep apnea (Hrubos-Strøm et al. 2011), yet approximately 70–80% of all cases are expected to be undiagnosed (Punjabi 2008). Studies show that sleep apnea sufferers are about two to three times more likely to be involved in traffic accidents because of the severe sleep deprivation (McNicholas 2013). In conclusion, untreated sleep apnea can lead to serious health implications, and in severe cases, even death. With numbers as high as these, it is clear that sleep apnea has a severe impact on both the health of the individuals, as well as society as a whole. Early diagnosis is crucial to reverse the course of the disorder.

2.1.1 Types of Breathing Disruptions

A breathing disruption is classified as either an *apnea* or a *hypopnea.* An *apnea* (or apneic event) refers to a complete cessation of breath, whereas a *hypopnea* (or hypopneic event) refers to a period of shallow breathing. According to the scoring rules by the *American Academy of Sleep Medicine* (AASM) (Berry et al. 2012), a breathing stop has to be at least ten seconds in duration to be classified as an apnea. For a period of shallow breathing to be considered a hypopnea, there has to be a minimum of 30% reduction in airflow lasting a minimum of ten seconds, and either a $\geq$ 3% drop in oxygen saturation, or an arousal. An arousal means that the patient has awoken for three to fourteen seconds (i.e., an EEG signal shows waking activity).

2.2 Types of Sleep Apnea

There are three types of sleep apnea: *Obstructive Sleep Apnea* (OSA), *Central Sleep Apnea* (CSA), and a combination of the two, often referred to as *Mixed* or *Complex Sleep Apnea.* A brief description of these is given in the following subsections.

2.2.1 Obstructive Sleep Apnea

Obstructive Sleep Apnea (OSA) is the most common type of sleep apnea and is characterized by a physical blockage of the upper airways during sleep (ASAA 2018b). When the muscles supporting the upper airways relax too much, the airways collapse, which obstructs breathing. The blockage may either be complete or partial, which corresponds to apneic and hypopneic events, respectively. During an obstruction, the body will automatically increase respiratory effort, which often results in loud snoring as air is forced through the blockage. OSA can affect anyone and is not restricted to people of a certain age, sex, weight, etc. On the other hand, OSA seems to be a little more prevalent in overweight, middle-aged males who sleep on their back. A very effective treatment for OSA is *continuous positive airway pressure* (CPAP) (MayoClinic 2018a), which is a face mask device generating positive airway pressure which keeps the airways open during sleep. CPAP requires continuous use, and can, therefore, feel burdensome for many people. An illustration of airway blockage is shown in Figure 2.1. In this particular illustration, the tongue falls back into the throat, completely blocking the airways.

Risk Factors

The risk of OSA is directly linked to the muscle tone in the throat, and how narrow the airways are. Weaker muscles are less able to support the soft tissue in the throat and keep the airways open, while narrower airways increase the impact of obstructions. Several physiological and environmental factors can influence these aspects, and thus, also the risk of having OSA (Alaska Sleep Clinic 2015). As people age, they tend to lose muscle tone and definition, resulting in weaker muscles in the throat. Alcohol is another factor which can cause the muscles to relax too much, resulting in an obstruction. OSA is also in many cases directly linked to obesity. Obese people are much more likely to have fatty tissue built up in the throat and neck, which may result in narrower airways and more tissue for the muscles to support. Another risk factor is genetic predisposition. Some people may naturally have a larger tongue or more narrow airways.

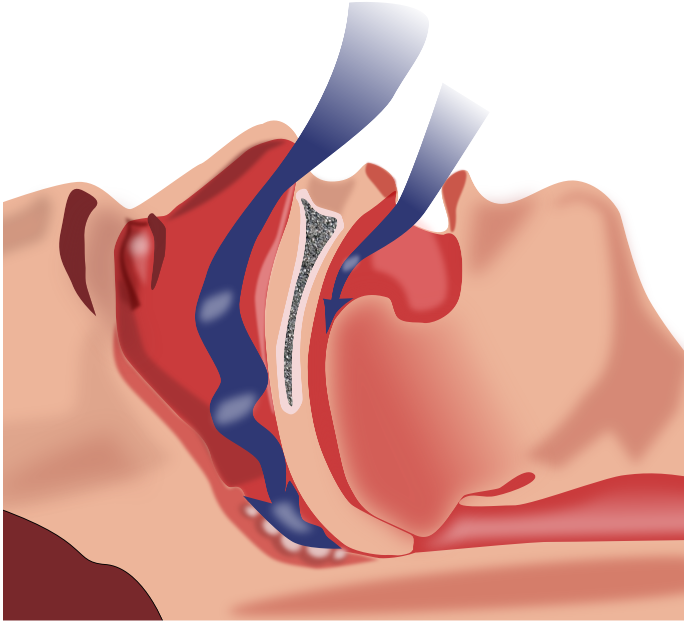


Figure 2.1: Illustration of obstructed airways (OSA) (M'henni 2010)

2.2.2 Central Sleep Apnea

In cases of *Central Sleep Apnea* (CSA), no physical blockage or obstruction are causing the disorder (ASAA 2018a). Instead, it is the brain temporarily failing to signal the muscles responsible for breathing, to breathe. When the oxygen saturation in the blood becomes too low, the brain will force an awakening to restore normal oxygen levels. As a result of the failed signaling, no apparent respiratory effort is present, and thus, only apneic events can occur. It is believed that CSA cases constitute of less than 20% of all sleep apnea cases, which makes it far less common compared to OSA. Since there is no physical blockage in CSA cases, treatment with CPAP is generally less effective. Instead, alternative treatments include the use of more expensive breathing machines, so-called *assisted ventilation devices* (Sleep Apnea Guide 2018).

Risk Factors

CSA is very often a comorbid disorder (ASAA 2018a), which means that people who are already very ill from other diseases are at higher risk of having CSA. This includes people with heart disorders and people who have had strokes or brain tumors. In addition, people who are male, over 65 years old, or using opioids, are also at higher risk of having CSA (Alaska Sleep Clinic 2015).

2.2.3 Complex Sleep Apnea

Mixed or *Complex Sleep Apnea* occurs when a subject is exhibiting symptoms of both OSA and CSA. This form of sleep apnea was first discovered by Morgenthaler et al. (2006), which makes it a more recent discovery compared to OSA and CSA. In this study, they observe that when some people with OSA receive CPAP treatment, they suddenly show symptoms of CSA. The underlying cause, however, remains unknown. The scoring rules by the AASM (Berry et al. 2012) also include a definition of *mixed apneas*. A mixed apneic event in adults is classified as an apnea which starts as a central apnea (i.e., no respiratory effort), and then turns into an obstructive apnea. For example, the first ten seconds of the event may show no respiratory effort at all (i.e., central), while the last ten seconds do show respiratory effort (i.e., obstructive). For pediatric patients, the *central* portion of the event may be located at either the start or end of the event.

2.3 Symptoms

As sleep apnea presents itself while the subject is sleeping, it may easily go unnoticed. In fact, the most common symptom is daytime sleepiness. Feeling tired can be normal for many people for various reasons, and maybe even more so for long-time sufferers of sleep apnea. A study by Van Dongen et al. (2003) suggests that people, in general, are mostly unaware of being sleep deprived, which further substantiates the claim that people often are unaware of having the disorder. Obstructive sleep apnea is caused by an obstruction of the airways, which very often results in loud snoring during sleep. Snoring is, however, usually not noticed by the sufferers themselves, but more often by a bedside partner. The first suspicion of having sleep apnea, therefore, often comes from a bedside partner noticing either loud snoring, breathing stops, or shortness of breath. One must, however, note that not everyone that snores has the disorder, nor does everyone with the disorder snore (e.g., subjects suffering from CSA do not snore). Snoring is, nonetheless, one of the most common first suspicions of having the disorder. Other symptoms of sleep apnea include *lack of concentration, hypertension (high blood pressure), memory loss, mood swings, headaches*, and *night sweats* (amongst others) (MayoClinic 2015). These symptoms are often easily dismissible as other causes, for example, a poor lifestyle, and as such, sleep apnea can be hard to suspect.

2.4 Diagnostic Tools

Sleep apnea does not leave any traces in the body that, afterward, can be detected and directly linked to the disorder. This means that blood tests or other tests alike cannot be used to diagnose sleep apnea. Instead, a sleep apnea diagnosis requires the subject to undertake a *sleep study*. In other words, the nocturnal events associated with sleep apnea have to be captured as they are happening at night, using various physiological sensors. The severity of the disorder is often determined based on the average number of apneic/hypopneic events recorded per hour, and this metric is known as the *Apnea-Hypopnea Index* (AHI). In addition, the severity of the disorder is often accompanied by other metrics as well, for example, the *Oxygen Desaturation Index* (ODI), and *Respiratory Disturbance Index* (RDI). In this section, we give an overview of the different severity ratings, physiological signals, and diagnostic tools used to diagnose sleep apnea.

2.4.1 Severity Rating

The most commonly used metrics for determining the severity of sleep apnea are defined as follows:

The *Apnea-Hypopnea Index* (AHI) is scored as the average number of apneic and hypopneic events per hour of sleep. This metric is mandatory for the diagnosis of sleep apnea according to the AASM (Berry et al. 2012) and is, therefore, widely used. The score of this metric is separated into four severity classes, *none/minimal*, *mild*, *moderate*, and *severe*, which are classified as follows (Harvard 2011):

- None/Minimal: AHI < 5 per hour
- Mild: AHI ≥ 5, but < 15 per hour
- Moderate: AHI ≥ 15, but < 30 per hour
- Severe: AHI ≥ 30 per hour

The *Oxygen Desaturation Index* (ODI) is an optional metric scored as the average number of oxygen desaturations per hour. According to the AASM (Berry et al. 2012), ODI is defined as:

$$ODI = \geq 3\% \ arterial\ oxygen\ desaturations\ /\ hour \tag{2.1}$$

The *Respiratory Disturbance Index* (RDI) is also optional and is defined by the AASM (Berry et al. 2012) as a composite metric. RDI is scored as the AHI plus the number of *respiratory effort related arousals* (RERAs) per hour:

$$RDI = AHI + RERA\ index \tag{2.2}$$

2.4.2 Physiological Signals

There is a wide range of physiological signals that can be used to diagnose sleep apnea (Tripathi 2008). In this section, we describe the most common ones, what they record, and how they are used.

Respiratory Sensors

Respiratory sensors are one of the most essential kinds of sensors for sleep apnea monitoring. It is, after all, the presence disrupted breathing that defines sleep apnea. These sensors are split into two different kinds, monitoring either *airflow* or *respiratory effort*. Airflow sensors monitor the flow of air as inhaled and exhaled through the nose and mouth. According to Berry et al. (2012), the gold standard sensor for monitoring airflow is a *pneumotachograph*, which is usually a mask placed over the mouth and nose. There are many types of pneumotachographs available, and one of which uses a turbine which rotates by the force of airflow. A pneumotachograph, however, is often somewhat large and bulky, which makes it unsuited for use in sleep studies. Other sensors monitoring airflow include a *nasal pressure transducer* and an *oronasal thermal sensor*. Both of these are recommended by the AASM (Berry et al. 2012) for apnea and hypopnea monitoring. A nasal pressure transducer is a small tube placed just below the nostrils, which measures airflow by the change in *pressure* of air as inhaled and exhaled through the nose. An oronasal thermal sensor is a small thermal sensor placed between the nose and mouth, which detects the presence of flow by the change of temperature as warmed air is exhaled. This thermal sensor, however, is not proportional to the actual airflow, and thus only detects the *presence* of flow.

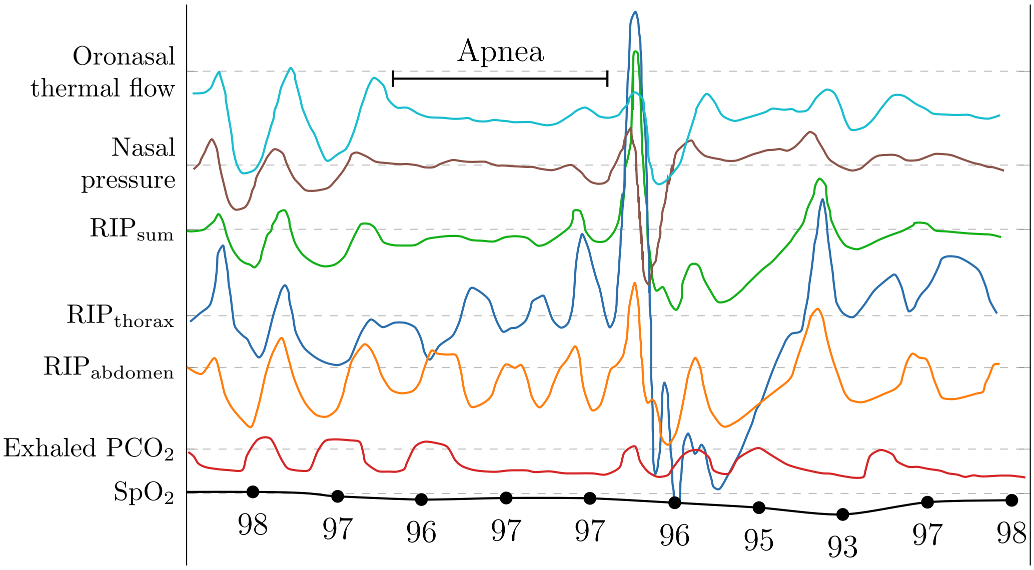


Figure 2.2: Respiratory signals during an obstructive apnea (Berry et al. 2012)

Respiratory effort sensors monitor the physical effort associated with respiration and are mainly used to separate obstructive from central events. The gold standard for monitoring respiratory effort is *Esophageal Manometry*, which is a tube inserted into the esophagus (i.e., a tube inserted into the throat stretching down to the stomach) (Berry et al. 2012). This kind of sensor is very uncomfortable and invasive for the subjects, and thus rarely used in practice. Other kinds of sensors include *respiratory inductance plethysmography* (RIP), *piezoelectric belts* (PZT), *impedance plethysmography* (IP), *polyvinylidene fluoride* (PVDF), and strain gauges (amongst others). Only the RIP and PVDF type sensors are recommended by the AASM (Berry et al. 2012). All of these sensors are regarded as noninvasive and measure the expansion and contraction of the thorax (chest) and abdomen associated with breathing. The RIP, PZT, PVDF, and strain gauges sensors, are belts strapped around the thorax and abdomen of the subject (often atop of clothing), whereas the IP sensor uses electrodes attached directly to the subject's skin. The advantage of RIP is that it measures the change of the total circumference of the belt. Contrary, the sensor part of a PZT belt, for example, spans only a small area of the full belt circumference, which means that it may or may not reflect the actual circumference of the belt (e.g., it may become trapped by lying on the sensor). The RIP, PZT, strain-gauges, and IP type sensors are described in more detail in Section 3.5.1. Even though these kinds of sensors are mainly used for monitoring respiratory effort, they can also be used to indirectly measure the inhaled and exhaled airflow/volume. It was first documented by Konno and Mead (1967) that the sum of the movement from the thorax and abdomen can be used to derive a semiquantitative estimate of lung volume. As a result, the AASM also recommends the RIP and PVDF sensors to detect breathing disruptions, and not only respiratory effort. We describe this in more detail in Section 3.5.

An example showing what these kinds of signals look like during an apneic event can be seen in Figure 2.2. As illustrated, the signal from the thorax and abdomen (RIP$_{thorax}$ and RIP$_{abdomen}$) show a presence of respiratory effort during the event, which classifies it as *obstructive*. If this event were of central type, the RIP$_{thorax}$ and RIP$_{abdomen}$ signals would flatline during the event (i.e., show no respiratory effort).

Pulse Oximetry

Another essential sensor is a *pulse oximeter.* A pulse oximeter is a small clip attached to either the fingertip, toe, or earlobe, and is used to measure the oxygen saturation of the blood. Apneic or hypopneic events are only significant when the oxygen saturation is affected to a certain extent (i.e., $\geq 3\%$ drop). The cells in the blood that carry oxygen are known as *hemoglobin.* Oxygen saturation is a measure of the proportion of oxygenated hemoglobin in the blood, and a value of 95–100% is considered normal for healthy people. An oxygen saturation level of 90%, 89–80%, and below 80%, are considered mild, moderate, and severe levels, respectively (Harvard 2011). A pulse oximeter consists of two different wavelengths of light, red and infrared. Oxygenated and non-oxygenated hemoglobin absorb these two wavelengths differently, and their ratio of absorption is used to derive the oxygen saturation in the blood (HEW 2018). The oxygen saturation derived from a pulse oximeter is referred to as SpO_2 and is usually calculated as an average over a *time window*, which makes it a *delayed* signal. An example of this can be seen in Figure 2.2, where the desaturation caused by the apneic event is first visible in the SpO_2 signal *after* the event is over. SpO_2 is an indirect measure of oxygen saturation, and the value of a direct measurement (e.g., by a blood sample) is referred to as SaO_2.

In addition to measuring oxygen saturation, a pulse oximeter can also be used to measure a wide range of other physiological signals. This includes, for example, the pulse rate, blood pressure (Talke et al. 1990), and changes in blood volume (photoplethysmograph) (Cannesson et al. 2008). Which of these signals a given pulse oximeter supports vary, with SpO_2 and pulse rate being the most commonly available signals.

Electroencephalography (EEG)

EEG measures the electrical activity of the brain using electrodes attached to the scalp (MayoClinic 2014). The brain emits different types of electrical activity depending on which state it is in, for example, delta waves ($\leq 3\ Hz$) during deep sleep and beta waves ($\geq 12\ Hz$) during periods of concentration (Brainworks 2018). EEG is mainly used in sleep apnea monitoring to determine when the subject is awake and asleep, which are further used to detect arousal associated with hypopneic events. In fact, EEG is the only reliable way of determining if the subject is asleep or awake. EEG is also used to detect which stage of sleep the subject is in and how much time the subject spends in each stage of sleep.

Electrocardiography (ECG)

ECG measures the electrical activity of the heart using electrodes attached around the thorax (and sometimes other places) (MayoClinic 2018b). ECG captures heartbeats represented by the waveform shown in Figure 2.3, from which many physiological features can be extracted. For example, the heart rate (beats per minute) can be determined by counting the number of QRS-complexes per minute. With respect to sleep apnea, it has been reported that variations in the RR-interval are associated with apneic/hypopneic episodes (Almazaydeh et al. 2012). The RR-interval is the time interval between two consecutive R-peaks in the signal. In contrast to respiratory signals, the ECG signal does not reflect any clear visual indications of apneic/hypopneic episodes. Such indications are

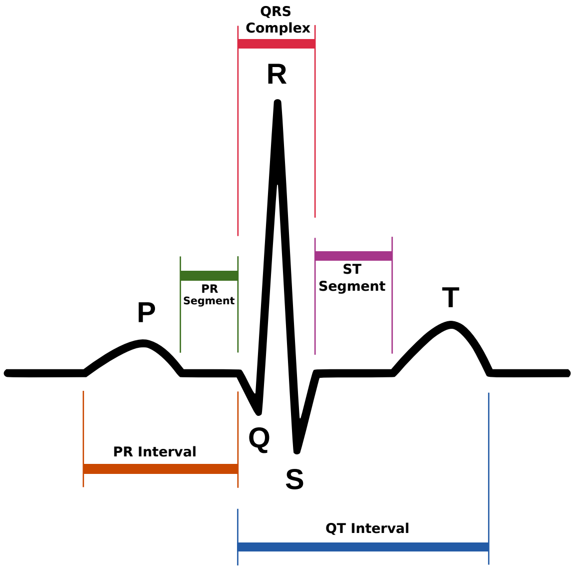


Figure 2.3: Waveform of a heartbeat from ECG (Atkielski 2007)

more often derived from various physiological features such as the RR-interval or heart rate (amongst others).

Electromyography (EMG)

EMG measures the electrical activity produced by muscles using electrodes attached to the skin. During sleep studies, two types of EMG are often recorded, chin EMG, and limb EMG (Tripathi 2008). Chin EMG is used to measure how the muscle tone of the throat and chin changes throughout the night and between different stages of sleep (Houston Sleep 2018). Limb EMG is often used to detect *restless leg syndrome* and *periodic leg movements of sleep* (PLMS).

Electrooculography (EOG)

EOG measures the movement of the eyes. In sleep studies, EOG is often used in combination with EEG to better determine the stage of sleep the subject is in (Estrada et al. 2006). During *Rapid Eye Movement* (REM) sleep, the brain waves recorded by the EEG are almost identical to that of wakefulness, which makes it hard to distinguish between the two. However, the eyes move sporadically and rapidly from side to side during REM sleep (hence the name), and thus, EOG is used alongside EEG to better distinguish between wakefulness and sleep.

Other Signals

In addition to the aforementioned signals, other important signals for use in sleep studies include a *microphone* and a *body position* sensor. The sound recorded by the microphone is used to determine if the subject is snoring, which can further be used to classify obstructive sleep apnea. A body position sensor is used because sleeping on the side rather than on the back reduces the number of apneic/hypopneic events drastically for some people (Katz and Dinner 1992) and (George et al. 1988). If, for example, a subject is sleeping on the side throughout the sleep study, they may not receive a diagnosis even though they do suffer from sleep apnea (i.e., AHI is lower than it should be). Capturing body position can, therefore, yield useful information for practitioners, for example, indicating that the subject should undertake an additional session.

Sampling Rates

The recommended sampling rates for the aforementioned signals during a sleep study can be seen in Table 2.1 (Tripathi 2008).

Signal	**Desirable**	**Minimal**
EEG	500 Hz	200 Hz
ECG	500 Hz	200 Hz
EMG	500 Hz	100 Hz
EOG	500 Hz	200 Hz
Airflow	100 Hz	25 Hz
Oximetry	25 Hz	10 Hz
Nasal Pressure	100 Hz	25 Hz
Esophageal Pressure	100 Hz	25 Hz
Body Position	1 Hz	1 Hz
Snoring Sounds	500 Hz	200 Hz
Rib Cage and Abdominal Movements	100 Hz	25 Hz

Table 2.1: Recommended sampling rates for the different signals (Tripathi 2008)

2.4.3 Types of Sleep Monitors

There is a wide range of different sleep monitors available for use in sleep studies, but not all of them employ the same set of physiological sensors. Additionally, the setting in which the sleep monitor is used also varies. For example, some monitors require being used in a sleep laboratory with continuous oversight of a trained sleep technologist, while others can be used at home without guidance. As a result, four different classes of sleep monitors have been defined. A sleep monitor may be classified as either Type I, II, III, or IV (CleveMed 2018), depending on what kind of sensors it employs and the setting in which it is being used. There are multiple definitions of these types, and the definition given by the Center for Medicare and Medical Services (CSM) is as follows:

- **Type I**
  A Type I monitor is required to be used in a sleep laboratory and must be manually monitored throughout the night by a sleep technologist. It must at least include the following signals: EEG, EOG, ECG, chin EMG, limb EMG, respiratory effort at thorax and abdomen, nasal airflow, and pulse oximetry.

- **Type II**
  A Type II monitor can be performed outside of a sleep laboratory without the oversight of a sleep technologist. The required signals are the same as for a Type I monitor.

- **Type III**
  A Type III monitor can be used at home without guidance and must include signals for airflow and respiratory effort, ECG/heart rate, and oxygen saturation.

- **Type IV**
  A Type IV monitor can be used at home without guidance. The required signals are not strictly defined, but it must have a minimum of three different signals, and these signals must allow for direct calculation of an AHI or RDI score (which is either airflow or thoracobdominal movement).

2.4.4 Polysomnography

The traditional way of diagnosing sleep apnea is by performing a sleep study known as *polysomnography* (which is a Type I monitor). Polysomnography requires the subject to spend the night in a sleep laboratory with a wide range of different physiological sensors attached to the body (see Figure 2.4). In addition, the recorded signals must be manually monitored throughout the night by trained medical personnel. Typical polysomnography includes signals for EEG, ECG, chin EMG, limb EMG, EOG, respiratory effort from the thorax and abdomen, nasal airflow, pulse oximetry, body position, and a microphone for snoring sounds (Tripathi 2008). Polysomnography is not specifically designed to diagnose sleep apnea but is widely used for many different kinds of sleep disorders. Sleeping in an artificial and unfamiliar environment with this many sensors attached to the body can for many people feel uncomfortable. It is not uncommon for people to not be able to fall asleep at all during polysomnography, which renders the results useless. In addition, this kind of sleep study is very resource demanding as it requires both expensive equipment, a suited laboratory, and trained medical personnel to manually monitor and analyze the results.

2.4.5 Portable Devices

As polysomnography is both very impractical, resource demanding, and uncomfortable for the subjects, a range of different portable devices have been developed. Portable devices, however, often have a much more limited set of sensors compared to traditional polysomnography. Some only have one sensor, whereas others have a few more. While most of these devices can be used at home without the help of trained personnel, the recorded signals often require being manually evaluated by trained personnel before an eventual diagnosis can be made. Many of these portable devices provide the ability to automatically score apneic/hypopneic events and calculate the severity rating of the disorder. Automatic scoring cannot give a definitive diagnosis, but only an indication. A few examples of portable devices are listed below.

- **ApneaLink Plus**
  ApneaLink Plus is a portable type III sleep monitor designed to be used at home without the help of trained personnel (ResMed 2018). It records signals for respiratory effort from the thorax, oxygen saturation, heart rate, and nasal flow. The included software scores apneic/hypopneic events automatically, along with providing a severity rating of the disorder.

- **Shimmer**
  Shimmer is a portable physiological platform (Shimmer 2018a). There is a range

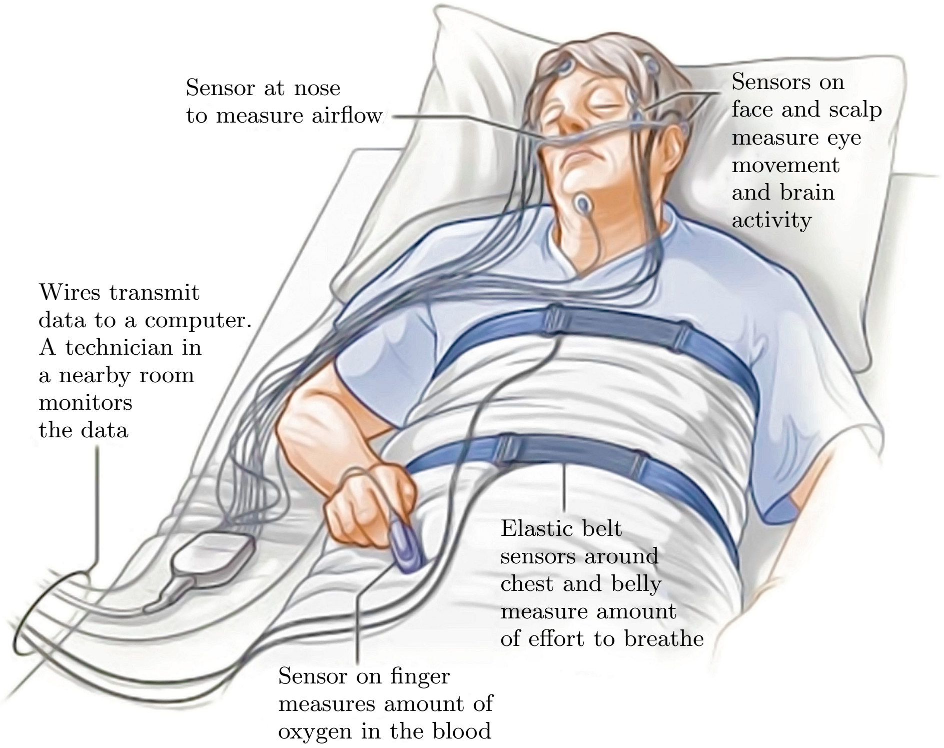

Sensor at nose to measure airflow
Sensors on face and scalp measure eye movement and brain activity
Wires transmit data to a computer. A technician in a nearby room monitors the data
Elastic belt sensors around chest and belly measure amount of effort to breathe
Sensor on finger measures amount of oxygen in the blood

Figure 2.4: Illustration of traditional polysomnography (NIH 2013)

of different sensor configurations available for Shimmer, with the most extensive configuration including sensors for ECG, EMG, respiratory effort from the thorax, and an accelerometer. Shimmer is not explicitly designed as a sleep monitor, but it can be used as one. Due to the lack of nasal airflow and oxygen saturation, it is classified as a type IV monitor.

- **NOX T3**
  NOX T3 is a complete medical grade respiratory type III sleep monitor made by NOX Medical (2018). The sensors supported by NOX T3 are dual thoracoabdominal respiratory effort belts (both RIP and PZT types), ECG, nasal pressure, pulse oximeter, accelerometer, snore sensor, and more. This device is widely used in hospitals and sleep centers for the diagnosis of sleep-related disorders around the world. Apneic/hypopneic events are scored automatically by the included software, and the AHI and ODI severity ratings are provided.

2.4.6 Questionnaires

As a part of the initial risk assessment evaluation for sleep apnea, several questionnaires have been developed. The most commonly used ones are known as the *Epworth Sleepiness Scale* (ESS) (Johns 1991), *G.A.S.P.* (Mazeika 2005), *STOP-BANG* (Shahid et al. 2011), and *Berlin* (Ahmadi et al. 2008). The number of questions in these questionnaires vary from five (in G.A.S.P.) to ten (in Berlin) and include questions on subjects such as overall sleepiness, snoring, weight, age, neck size, witnessed apneic episodes, and hypertension (amongst others). It has been shown that questionnaires are a useful method of assessing the risk of having OSA, but may be suboptimal when applied to the general population (Hrubos-Strøm et al. 2011).

2.5 Discussion and Conclusions

Sleep apnea is both hard to suspect, uncomfortable to diagnose, and very resource demanding. As a result, the threshold for a potential patient to perform the first step towards a diagnosis is currently too high, which makes it a severely under-diagnosed disorder. The consequences sleep apnea has on both the individuals as well as society as a whole, makes it crucial to decrease the number of undiagnosed cases. During sleep apnea monitoring, the most important parameters of interest are:

- **Airflow**
  The presence of airflow is *the* most important parameter with respect to sleep apnea monitoring. Sleep apnea is, after all, characterized as a disruption of airflow. Airflow can either be measured directly at the nose and mouth or indirectly by the expansion/contraction of the thorax and abdomen.

- **Respiratory Effort**
  To classify an event as either obstructive or central, recording respiratory effort is required. The gold standard for monitoring respiratory effort is with *esophageal manometry*, but the more widely used technique is to externally monitor the expansion/contraction of the thorax and abdomen (e.g., RIP belts).

- **Sleep and Wakefulness**
  The only reliable way of knowing if a subject is asleep or awake is with the use of an EEG. EOG is used alongside EEG to better distinguish between wakefulness and REM sleep. These signals are further used to monitor arousals associated with breathing disruptions.

- **Effect of Airflow Disruption**
  Oxygen saturation and ECG are both used to measure the *effect* the loss of oxygen has on the body, rather than the disruption of airflow directly.

Chapter 3

Measuring Data Quality

This chapter presents an overview of how the quality of data can be measured, or more precisely, quantified. The emphasis lies primarily on signals from respiratory effort sensors, but the methods can be applied to other types of data as well. We begin in Section 3.1 with a brief introduction to what it means to measure the quality of data, along with various types of measurement scales and data quality dimensions. We continue in Section 3.2 with a brief presentation of what physiological time series data is, how the signals from different sensors can be synchronized, and which data quality dimensions that are relevant for physiological time series data. In Section 3.3 we present some examples of signal quality indicators for pulse oximeters and ECGs, followed by various commonly used accuracy metrics in Section 3.4. Next, we describe respiratory effort sensors in more detail in Section 3.5, and finally, conclude the chapter in Section 3.6.

3.1 Introduction

The definition of *quality* is given by the Oxford English Dictionary as *the standard or nature of something as measured against other things of a similar kind; the degree of excellence possessed by a thing* (OED 2018a). This definition shows that the need to describe the quality of an entity (or some *thing*) is primarily derived from the desire to answer two questions: (1) how such an entity *compares* to other related entities, and (2) how the entity *performs* at a specific task. The *entity* may in our case be any kind of data that can have quality, such as the data in a database, the signal from a physiological sensor, etc. The answers to these two questions are not mutually exclusive. In fact, the answer to the second question can also be used to answer the first question, however, not the other way around. A *metric* can be defined as the *unit of measurement*, and in other words, *what* is being measured. Quality in itself can, in fact, be considered a metric, but without a clear and concise definition of precisely *what* is being measured, such a metric does not yield any valuable information.

There are many ways to measure the quality of an entity, and a preliminary method is often a subjective evaluation represented by metrics such as *bad, ok, good, better, best.* The main drawback with subjective evaluations like these is that they in the best case only yield very rough estimates to the above questions, lack a lot of information, and can vary significantly from person to person and from situation to situation. For example, given a physiological sensor with a quality rated as *ok.* It is impossible to know exactly *how* good the sensor may be, or even if it is good enough for a specific task. That kind of information is just not available. The concept of *quantifying* the quality is used to reduce this subjective influence while increasing the accuracy and amount of information in the given metric. For example, in the case of physiological sensors, representing the accuracy as ± 6% yields a lot more information compared to what *ok* does. It is clearly defined exactly *how* accurate it is, and its accuracy can even be compared directly to other sensors.

The important concept is, however, not quantification per se, but rather the *scale of measurement* the given metric is based on. In other words, the measurement scale of a metric defines *how* the metric can be compared and what mathematical operations it supports. The different scales we discuss further are known as *nominal, ordinal, interval*, and *ratio*, where each scale is an extension of the preceding scales. The supported mathematical operations for comparability for these scales are given as *equality/inequality* $(=, \neq)$, *rank-ordering* $(<, >)$, *difference* $(+, -)$, and *ratio* $(*, /)$, respectively. The interval scale is, in fact, the first of these scales that is quantitative because information about the difference between each value is clearly defined. As the ratio scale extends the interval scale, it too defines a quantitative measurement. The exact details of these measurement scales are described in more detail in Section 3.1.1.

Merely representing the quality by a number is, however, not the same as *quantifying* the quality. The quality can be represented by numerical symbols without giving meaning to their numerical values. Take the school grades *1–6* as an example. One cannot say definitively that the grade *4* is twice as good as the grade *2* because the grades are usually given based on a teacher's subjective evaluation. One knows which grade is considered better, but not exactly *how* much better one grade is compared to another. In this specific case, the measurement scale is ordinal because only the rank-order of the possible values is defined. Defining exactly what measurement scale the metric is based on is, therefore, required for the quality of two related entities to be comparable.

Defining *quality* itself as a metric is vague and imprecise because what quality means depends very much on the situation and context in which it is defined. Additionally, there needs to be something concretely defined to count if a quantitative measurement is the goal. For example, a student's performance on a simple math test is trivial to quantify. One may simply use the number of correctly answered questions as the metric for performance (or quality). This measurement is even ratio scaled because a student with a score of *20* did twice as good as a student with a score of *10.* A quantitative metric for a student's performance on an English essay, however, is not nearly as trivial to define. One may define the number of grammar mistakes as the quantitative metric, but grammar is yet only one (smaller) aspect of the overall quality of an essay. In this case, the overall quality consists of multiple *dimensions*, where grammar may be one, content another, and correct use of the genre a third. Regarding data quality, many different dimensions can be defined. The quality of a database can, for example, be measured based on how *complete* it is (i.e., how much information it is lacking), how *accurate* the data in the database is, and how *up-to-date* the data in the database is. How precisely one is able to define such dimensions determines what measurement scale the metric can be based on and whether or not it is quantitative. Several top-level generic data quality dimensions have already been defined, which we describe further in Section 3.1.2.

To summarize, there are primarily two factors that contribute to how well the aforementioned questions can be answered. The first being the measurement scale of the metric, which describes how one might compare the quality with other related entities. Second, a precise definition of *what* is being measured (i.e., dimension/metric) is required to determine an entity's performance at a given task. A perfect metric can answer both of these questions precisely; however, such a metric might not exist or may be infeasible to achieve in certain situations.

3.1.1 Measurement Scales

The scale of a metric defines its *comparability*, and in other words, what mathematical and statistical operations it supports. The measurement scales we describe further were first introduced by Stevens (1946) as *nominal*, *ordinal*, *interval*, and *ratio*. There have later been defined other scales as well, but these cover mostly edge cases and are not relevant to our situation. The mathematical operations for comparability for these scales include *equality/inequality* $(=, \neq)$, *rank-order* $(<, >)$, *difference* $(+, -)$, and *ratio* $(*, /)$. In addition, their support for statistical measures of central tendency also varies (e.g., *mode, median, mean, etc.*). These measurement scales are defined as follows:

Nominal The *nominal* scale is at the bottom of the measurement scales and contains the least information regarding how two entities compare. When a metric is nominally scaled, only the equality/inequality of the values can be compared. It does, in other words, just support these mathematical operations for comparability: $=, \neq$. It is not possible to determine which entity is better or which is worse at all. A nominal measurement is, in other words, a measure of *class* and can only be used for classification purposes. The supported measure of central tendency is, thus, only *mode*.

Ordinal The *ordinal* scale extends the nominal scale by adding information about the rank-order of the possible values. The order becomes meaningful and which entity is considered *better* is, thus, known. However, one still cannot know *how* much better one value is compared to another because the difference between the values is yet unknown. The only supported mathematical operations for comparability are: $>, <, =, \neq$. Taking the grades A-F as an example. One knows that the grade A is better than the grade B, but the difference between the grades A and B might be the same as the difference between B and D for all one knows. An ordinal scaled measurement supports both *mode* and *median* as measures of central tendency. Because the difference between the values is missing, calculating the arithmetic mean or standard deviation are meaningless.

Interval The *interval* scale extends the ordinal scale by adding information about the *difference* between the possible values. The interval scale is, in fact, the first of these scales that defines a quantitative measurement. With an interval scale, the difference between the values is clearly defined, but a clear definition of zero is, however, still lacking. This means that one cannot take ratios and conclude that anything is twice as good or bad as anything else. The supported mathematical operations for comparability are: $+, -, >, <, =, \neq$. An example of this measurement scale is the degrees Celsius.

The temperature change between 10℃ and 20℃ is the same as the temperature change between 30℃ and 40℃ (i.e., 10℃ warmer in each case). However, the temperature 40℃ is not twice as hot as 20℃. If it were, then that would imply that 0℃ is the total absence of temperature, which is not the case. An interval scaled measurement supports all the aforementioned measures of central tendency along with the arithmetic mean and standard deviation.

Ratio A *ratio* scaled measurement has the same properties as an interval scaled measurement, but with the addition of having a clear definition of zero. This means that one can take ratios and conclude that a sensor with an accuracy score of 10 is twice as accurate as a sensor that only has a score of 5. The supported operations for comparability of a ratio scaled metric are: $*, /, +, -, >, <, =, \neq$, and all the aforementioned measures of central tendency are supported.

In other words, the ratio scale extends the interval scale, which in turn extends the ordinal scale, which extends the nominal scale. As such, it becomes clear that the ratio scale is the optimal scale in terms of properties, but it is, on the other hand, not possible to design a metric which is ratio scaled in all situations. An example of this is with user satisfaction, which is inherently subjective. Given a metric with the values *Not Satisfied*, *Quite Satisfied* and *Very Satisfied*, it is clear which of these values that are considered "better," but not the difference between them. In conclusion, the quality of an entity must at least be represented as an interval scaled metric to be considered quantitative. The closer to a ratio scaled metric the better, but not all situations allow for it (or need it).

To determine which measurement scale a given metric is based on, one can check what transformation functions the measurement supports. A transformation function is a mathematical function which transforms the values of the measurement to other values. Such a function is supported if it does not change the measurement results according to the given scale. All the scales support multiplication and division by a constant. This changes the size of the units but does not alter the ratios, intervals, rank-orders, or the classes of the measurements. If the scale supports addition or subtraction, it cannot be ratio scaled because the values' relative ratios would change. If the scale supports squaring of the values, then it cannot be interval scaled. By squaring the values, the interval between the values changes, but their rank-order, however, remains intact. If the scale supports substituting one value for any other, then it cannot be ordinally scaled, as that would change the rank-order of the values. The nominal scale is the only possibility left should all the transformations fail.

3.1.2 Data Quality Dimensions

When quantifying or assessing the quality of data, it is common practice to separate and evaluate multiple smaller dimensions of that data. A Data Quality (DQ) dimension describes a feature of some data which can be measured, assessed, counted, etc. to determine the quality of the data. In other words, a DQ dimension can be a feature of a data item, a sensor signal, a record, a dataset or a database that can be measured to understand its quality. Askham et al. (2013) have defined and published six generic best practice dimensions for use in data quality assessment, known as *Completeness*, *Uniqueness*, *Timeliness*, *Validity*, *Accuracy*, and *Consistency*. The motivation behind this work is mainly to reduce confusion and uncertainty amongst practitioners when considering data quality. Previously, many practitioners have used different terms to describe the same DQ dimension. For example, the terms *Accuracy* and *Correctness* have often been used interchangeably to describe the same dimension. The standard data quality dimensions are defined as follows:

Completeness The *completeness* dimension describes the proportion of the actual stored data in relation to the potential 100% complete. In other words, completeness measures the absence of missing data in percentage (e.g., blank, null, or empty values). For example, given a database with the contact details of 200 employees where the e-mail addresses of nineteen of these employees are missing. The completeness of this dataset (of email addresses) is calculated to be $\frac{200-19}{200} = 0.905$, or in other words, 90.5% complete. The general formula can be seen in Equation 3.1, where $|entities|$ is the number of entities representing the potential 100% complete, and $|entities_{dataset}|$ is the number of entities in the dataset.

$$Completeness = \frac{|entities_{dataset}|}{|entities|} \times 100\% \tag{3.1}$$

Uniqueness There should not be more instances of an entity in a given dataset than what is present in the real world. If there only exists one instance of an entity in the real world, then there should also just exist one instance of that entity per dataset. For example, if a student is recorded as both *Eissonhour* and *Eisenhower* in the school's database, the uniqueness dimension is not perfectly fulfilled. Uniqueness is given as a percentage and can be calculated after the formula in Equation 3.2, where $|entities_{dataset}|$ is the number of entities in the dataset and $|entities_{real\_world}|$ is the number of real-world entities being described by the given dataset.

$$Uniqueness = \frac{|entities_{real\_world}|}{|entities_{dataset}|} \times 100\% \tag{3.2}$$

Timeliness The *timeliness* dimension describes how *up-to-date* the data is. How this dimension is assessed varies between different contexts. It may be assessed by the proportion of data that is up-to-date, or it can be assessed by how old an entity is (i.e., continuous). For example, it is not unlikely for a student to change their telephone number. Thus, timeliness can be represented by the proportion of telephone number records in the dataset that are still up-to-date. The maximum age of an entity must be chosen depending on the context, and one may, for example, regard the maximum age of a student's telephone number to be one year. A telephone number is then regarded as out-of-date once it exceeds this age. The formula giving the proportion of up-to-date entities is given in Equation 3.3, where $|entities_{up\_to\_date}|$ is the number of up-to-date entities, and $|entities|$ is the total number of entities in the dataset.

$$Timeliness = \frac{|entities_{up\_to\_date}|}{|entities|} \times 100\% \tag{3.3}$$

Validity The *validity* dimension describes the degree to which the data is valid (e.g., by conforming to the intended syntax). For example, should a date be encoded as *DD/MM/YYYY* or *MM/DD/YYYY*? Can a postal code be a negative number? Can the heart rate of a human be negative? Validity is given as a percentage of how many of the entities in the dataset that are regarded as valid. The formula is given in Equation 3.4, where $|entities_{valid}|$ is the number of valid entities, and $|entities|$ is the total number of entities in the dataset.

$$Validity = \frac{|entities_{valid}|}{|entities|} \times 100\% \tag{3.4}$$

Accuracy The *accuracy* dimension describes the degree to which the recorded data accurately describes the real world entities it represents. In other words, the accuracy of an entity is the *distance* between what it *is*, and what it *should be*. For example, if the real name of a person is *Eissonhour* and the entity is stored as *Eisenhower*, the accuracy can be described as 60%. This is calculated using a domain-specific distance function. In this example, the distance is calculated by using the *Levenshtein edit distance* function, which represents the minimum number of changes required to transform one string into another. For the *Eissonhour*/*Eisenhower* example, the edit distance is four. Since the number of characters in the longest of the two strings is ten, the accuracy is calculated as $1-\frac{4}{10} = 60\%$. The accuracy may be represented as the proportion of *accurate* entities in a dataset or as an aggregation of the domain-specific distance function. The formula for the former is given in Equation 3.5, where $|entities_{accurate}|$ is the number of accurate entities, and $|entities|$ is the total number of entities in the dataset. An example formula for the latter is shown in Equation 3.6, where *entities* are all the entities in the dataset, *truth* their corresponding ground truth counterparts, and $d(x, y)$ a domain-specific distance function.

$$Accuracy = \frac{|entities_{accurate}|}{|entities|} \times 100\% \tag{3.5}$$

$$\begin{aligned} n &= |entities| \\ Accuracy &= \frac{1}{n}\sum_{i=0}^{n} d(entities_i, truth_i) \times 100\% \end{aligned} \tag{3.6}$$

Consistency The *consistency* dimension describes the percentage of entities stored in one dataset that are consistent with the same entities stored in another dataset. For example, an organization may have multiple datasets, and the home address of an employee should be equal to and stored in the same format across all these datasets. The general formula is shown in Equation 3.7, where $|entities_{consistent}|$ is the number of entities that are deemed consistent across datasets, and $|entities|$ is the total number of entities in the given dataset.

$$Consistency = \frac{|entities_{consistent}|}{|entities|} \tag{3.7}$$

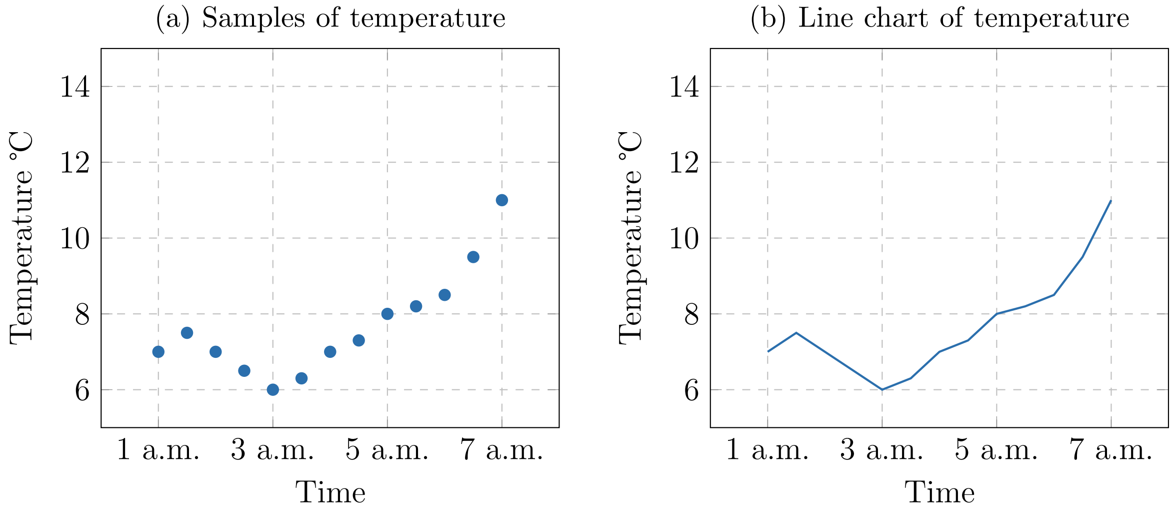


Figure 3.1: Example of time series

3.2 Physiological Time Series

A *time series* is, simply put, any variable y with an added time dimension x. The variable y changes value in relation to time, and in a time series, what the value of y is at a given point in time is indexed by the time dimension x. Take outdoor temperature as an example. The temperature may be 10°C at 4 a.m. and 23°C at 2 p.m. The temperature is represented by the y variable and time by the x variable. Both temperature and time are in reality continuous variables, which means that the number of values in the interval between any two values is infinite. Recording continuous variables is infeasible to do, and instead, discrete samples of both variables are captured, usually at equally spaced points in time. This is referred to as *sampling*, and the rate/frequency of the sampling is referred to as the *sampling rate* or *sampling frequency*. The sampling rate may be expressed in *Hertz*, e.g., 20 Hz means 20 samples every second, or as an interval, e.g., one sample every 50 milliseconds.

An example of a time series can be seen in Figure 3.1, with the discrete samples of the recording of temperature over time in Figure 3.1a. In this example, each sample is taken at a 30-minute interval. Because these are discrete samples of continuous variables, the values between any two samples are lost. These missing values may be partially restored through *interpolation*, which means *guessing* what the values may have been based on the nearby samples. One simple form of interpolation is called *linear interpolation*, which means to connect each adjacent sample with a straight line. An example of linear interpolation can be seen in Figure 3.1b. There are other algorithms for interpolation as well, for example, fitting a quadratic or cubic polynomial between each point. Interpolation of higher order makes the line between each point smoother compared to linear interpolation, and maybe more accurate depending on the type of data.

The signal from a physiological sensor, for example, an ECG, pulse oximeter, or a respiratory effort belt, is just a time series like the example in Figure 3.1. The actual signal from the sensors is represented by the y variable and the time at which the values are recorded by the x variable. However, the nature of the signals varies from sensor type to sensor type. For example, capturing ECG data (i.e., heartbeats) requires a somewhat high sampling rate. Even though a heartbeat may occur only once per second, a sampling rate of 1 Hz is not nearly high enough to capture the *shape* of the heartbeat (also referred to as the QRS-complex). Oxygen saturation, on the other hand, is a much slower process, which means that a sampling rate of 1 Hz may be enough.

3.2.1 Synchronizing Signals

Before any two signals can be compared, they must be synchronized. There are mainly two parts to synchronization: (1) the samples need to be captured at the same instant in time, and (2) the sensors' internal clocks must be synchronized. For example, given two arbitrary sensors with a sampling rate of 10 Hz. If these sensors are not started at exactly the same time, then none of the captured samples will ever be captured at the same instant in time. Every sample from one signal may end up right between two samples of the other signal. There is no way to perfectly synchronize the signals in software after they are captured, and the precision is determined solely by the sampling rate. When the sampling rate is 10 Hz, then the precision of the synchronization is within $\pm 50ms$. Interpolation may be used to increase the precision to a certain extent by upsampling the signals, but it will never be perfect. If the sensors, on the other hand, are started at the exact same time, the signals are still not synchronized if their internal clocks are not synchronized. In this case, a perfect synchronization can, in fact, be achieved in software. Both signals' y values are in this case already synchronized, and only their time dimension (x values) needs to be adjusted.

Starting the signal capture of two sensors at the exact same time is impossible to do manually. A *perfect* synchronization can, in other words, only be achieved through hardware methods. Some sensors do have support for hardware synchronization, which means that they exchange information about *when* to capture a sample, and synchronize their internal clocks. Hardware synchronization is, however, rarely supported, and especially not so with sensors from different manufacturers. Synchronization through software techniques is, therefore, the more common approach.

One way to perform synchronization in software is with the use of *cross-correlation* (Silva et al. 2015). Cross-correlation measures the similarity (covariance) between two time series for every possible displacement of one relative to the other. After calculating the similarity for every possible displacement, the displacement where the signals are most similar (i.e., covariance is maximum) is assumed to be the correct point of synchronization. The formula for cross-correlation is shown in Equation 3.8, where $\hat{y}$ is the first signal, y the other signal, and d^* the displacement index of the synchronization point.

$$d^* = \underset{d\, \in\, \mathbb{Z}}{\arg\max}\left(\sum_{i=-\infty}^{+\infty} \hat{y}[i]y[i+d] \right) \tag{3.8}$$

This particular definition of cross-correlation assumes that the sampling rates of the two signals are equal. Although, it can quite easily be adjusted to work with signals of different sampling rates as well by changing the indexing. The use of cross-correlation for synchronization comes with a requirement. It assumes that the signals are indeed most similar at the correct synchronization point. In the presence of, for example, noise, this may not be the case. Cross-correlation is, therefore, more suited for some signals compared to others. ECG and respiratory signals, for example, have the advantage that the heartbeats and breaths are somewhat distinct features of the signals. The oxygen saturation signal from a pulse oximeter, on the other hand, has less distinct features to base the synchronization on.

Equalize Sampling Rates

To equalize the sample rates of time series, one would either have to perform a downsampling, upsampling, or a combination of the two. An upsampling may be performed by interpolating the signal, and then pick new samples from the interpolated values. To avoid having to *guess* too many of the missing samples, it is often preferred to downsample rather than to upsample a signal. There are many ways to downsample a signal, and one of the more simpler ways is to remove every n^{th} sample from the signal. For example, to downsample a 1000 Hz time series down to 500 Hz, one removes every other sample. In cases where the original sampling rate is not divisible by the new sampling rate, a new sample may need to be inserted between two existing samples, in which case taking the mean value of the nearby samples is a common approach. Taking the minimum, maximum, or sum are also common approaches, and the best method depends on the type of signal. Downsampling is, however, often a part of a process known as *decimation*. By just downsampling a signal, there may still be traces of higher frequency bands left in the signal, which may cause *aliasing* (depending on the type of signal in question). The process of decimation, therefore, often consists of filtering out the higher frequency bands before the downsampling is performed.

3.2.2 Data Quality Dimensions

The standard DQ dimensions described in Section 3.1.2 are regarded as generic, and as such, most likely need to be customized to fit the intended context. In addition, all of the dimensions may not even be relevant in all situations. Some of the dimensions, like *uniqueness* and *consistency*, are mostly tailored for the semantic of a dataset or a database, and not so much for individual entities by themselves. In this project, we mainly want to measure the quality of physiological time series data. As such, the standard DQ dimensions relevance to physiological time series data are discussed further below.

Completeness In the context of physiological time series data, completeness can be seen as the difference between the actual length of the time series in relation to its intended length. For example, a user may be supposed to monitor respiratory signals for eight hours, but the recording only contains six hours of data. In this case, the data can be seen as 75% complete. In addition, the sensor may be taken off for a short period or repositioned in the middle of the session, which may corrupt the data. In this case, that data can either be seen as invalid or be removed. In case of the latter, the completeness is affected. Another example is with the use of wireless sensors (e.g., Bluetooth), which may cause the signal to drop or become weak at certain points. For instance, if the user goes to the bathroom in the middle of the session, depending on the range of the signal, the connection might drop. Now the question of whether this situation should impact the calculated signal quality or not arises. One may argue that as the patient was awake during the incident, and sleep apnea is monitored, the right thing to do may be to exclude the incident from the quality analysis altogether. This assumes, however, that a signal drop associated with an event like that can be distinguished from a signal drop happening as the patient is sleeping in bed (which *should* be taken into consideration).

Accuracy When measuring the accuracy of an entity, it needs to be compared against its real-world counterpart. In the context of physiological time series data, the accuracy of a signal is, simply put, the *distance* between it and either a ground truth (if available) or a gold standard signal. Defining *how* the distance between different signals should be calculated, however, is not a trivial task. The mere difference between the values of the two signals is one option, but that requires that both signals are in the same units of measurement to be meaningful. Moreover, the raw signal may contain parts that are irrelevant for the performance of a sensor in a given context. A raw signal comparison may, therefore, not reflect the sensor's actual performance. We describe a number of common statistical accuracy measures further in Section 3.4.

Validity The validity dimension can be used to represent which parts of a time series that are deemed valid and invalid, or in other words, the proportion of time the signal is valid. Determining *what* a valid signal is may seem trivial on the surface, but it can, in fact, be rather obscure. In its essence, the validity of a signal comes down to what the signal is *expected* to be (or behave) like in relation to what it *actually* is. For example, the oxygen saturation in the blood is restricted to the range 0–100%. Although, an oxygen saturation which is too low (e.g., less than 50%) is most certainly wrong if the patient is indeed still alive and in a normal condition. Another example is a flat-lined pulse. If a patient is alive, then a flat-lined pulse most certainly means that the signal is invalid. In contrast to accuracy, the measure of validity can be a purely intra-signal measure. Meaning that no additional ground truth/gold standard signal is required to measure the validity of a signal. As a result, the validity dimension is very suited for giving real-time quality feedback or indications to the user. *Validity* measures for sensor signals are also referred to as *signal quality indicators*. We describe further what such indicators look like and how they may be used, in Section 3.3.

Excluded Dimensions The *Uniqueness*, *Timeliness*, and *Consistency* dimensions are not relevant when assessing the quality of individual time series. Since we want to assess the quality of a given time series, and not how that time series is stored in a database, it is irrelevant whether it is unique/consistent or not. Physiological time series data will also not become out-of-date because it represents reality at the specific point in time in which it was captured.

3.3 Signal Quality Indicators

The accuracy of a sensor is measured by the distance between what it is and what it should be, based on a ground truth or gold standard signal. Validity, on the other hand, is a measure of what the signal is *expected* to look (or behave) like in relation to what it actually is. As a result, validity measures are rarely quantitative in the form of knowing exactly *how* valid the signal might be. It is more often based on a boolean threshold, either it is valid, or it is not. Validity measures are also known as *signal quality* *indicators*, which are various indicators in a signal that can be used to describe its validity. Whenever the signal takes on unexpected values or behavior, then it is suspected to be invalid. As mentioned, what a valid signal actually means is very dependent on the type of sensor in question. Therefore, we describe in this section several examples of signal quality indicators in the context of pulse oximetry and ECG signals. These examples are originally described in a patent by Baker and Richards (2005), and a brief description is given as follows:

Overlap Put shortly, a pulse oximeter uses two wavelengths of light, red and infrared, to determine the oxygen saturation in the blood. Hemoglobin in the blood carrying oxygen absorbs these wavelengths of light differently than hemoglobin not carrying oxygen, and their ratio of absorption is used to derive the oxygen saturation level. The *overlap* indicator determines the degree to which the signals from the two different wavelengths overlap. Overlap is an indirect measure of the extent to which the two wavelengths probe the same volume of tissue. If, for example, the sensor is misplaced or if there is dust/hair on the sensor, then the two wavelengths may not probe the same tissue. The more the various wavelengths differ, the more the quality is known to degrade. There are many algorithms to calculate overlap, and an example is given in Equation 3.9, where the summation can span from one to multiple seconds.

$$\begin{aligned} R &= \frac{In(Red_{max}/Red_{min})}{(IR_{max}/IR_{min})} \\ Overlap &= \frac{\sum min(IR_t - IR_{min}, (Red_t - Red_{min})/R)}{\sum IR_t - IR_{min}} \end{aligned} \tag{3.9}$$

Min-Max-Min The blood pressure between heartbeats is known as *diastolic* pressure, whereas the blood pressure during a heartbeat is known as *systolic* pressure. During a cardiac cycle (heartbeat), it is a known fact of the human physiology that it should take less time for the blood pressure go from minimum (diastolic) to maximum (systolic), than from maximum (systolic) and back to minimum (diastolic). If this is not the case for a signal, then there has to be a quality problem. The *min-max-min* indicator represents the time ratio between the duration of going from diastolic pressure to systolic pressure in relation to going from systolic pressure and back to diastolic pressure. This indicator can be calculated as shown in Equation 3.10, where *systole* is the time of systolic pressure during a cardiac cycle, $diastole_{start}$ is the time of diastolic pressure at the start of a cardiac cycle, and $diastole_{end}$ is the time of diastolic pressure at the end of a cardiac cycle. If x is more than or equal to 1, then there is a quality problem.

$$x = \frac{systole - diastole_{start}}{diastole_{end} - systole} \tag{3.10}$$

Path Length The *path length* indicator measures the frequency content of the signal relative to the pulse rate. If the frequency content of the signal is higher than the pulse rate, then that means that it is being affected by something other than the pulse itself. A high frequency means that something is changing in the measurement, and if it is not caused by the pulse rate, then it is most likely caused by physical movement of the sensor.

It is known that physical movement is a common error source of pulse oximeters. An example of how path length can be calculated is given in Equation 3.11.

$$PathLength = \frac{\sum_{i=0}^{Samples\_in\_Pulse-1} |IR_{t-i} - IR_{t-i-1}|}{Pulse_{max} - Pulse_{min}} \tag{3.11}$$

IR nAv This indicator measures the infrared (IR) light level of the sensor. A low infrared light level is often caused by misplacement or by placing the sensor over something other than skin (e.g., dust/hair). After normalizing the light level for the specific light source in the sensor, it can be determined if the light level is lower than it should be.

3.4 Accuracy Metrics

The accuracy dimension requires, as mentioned, a domain-specific distance function. Defining such a function is not trivial for all types of data. In this section, we describe some common statistical metrics and distance functions that can be used to determine the accuracy of an entity.

3.4.1 Distance

A distance function $d(x, y)$ is a function that satisfies the following requirements (Garcia-Molina et al. 2008, pp. 1125–1126):

1. $d(x, y) \geq 0$ for all points x and y
2. $d(x, y) = 0$ if and only if $x = y$
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, y) \leq d(x, z) + d(z, y)$ for any points x, y and z (triangle inequality)

> That is, the distance from a point to itself is 0, and the distance between any two different points is positive. The distance between points does not depend on which way you travel (symmetry), and it never reduces the distance if you force yourself to go through a particular third point (the triangle inequality).

One of the most common distance functions is known as the *Euclidean distance* (ibid.). This distance function is a measure of distance between points in a *Euclidean space*, or more simply put, an n-dimensional space. For a simple example, consider the 2-dimensional space shown in Figure 3.2a. The two points in this diagram have the coordinates (2,2) and (3,3), and their Euclidean distance is illustrated by the straight line connecting the two points. One may notice that for 2-dimensional points, this distance is actually the same as the length of the hypotenuse of a right-sided triangle following the *Pythagorean theorem.* The formula for Euclidean distance can be seen in Equation 3.12, where x and y are two points in an n-dimensional space.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.12}$$

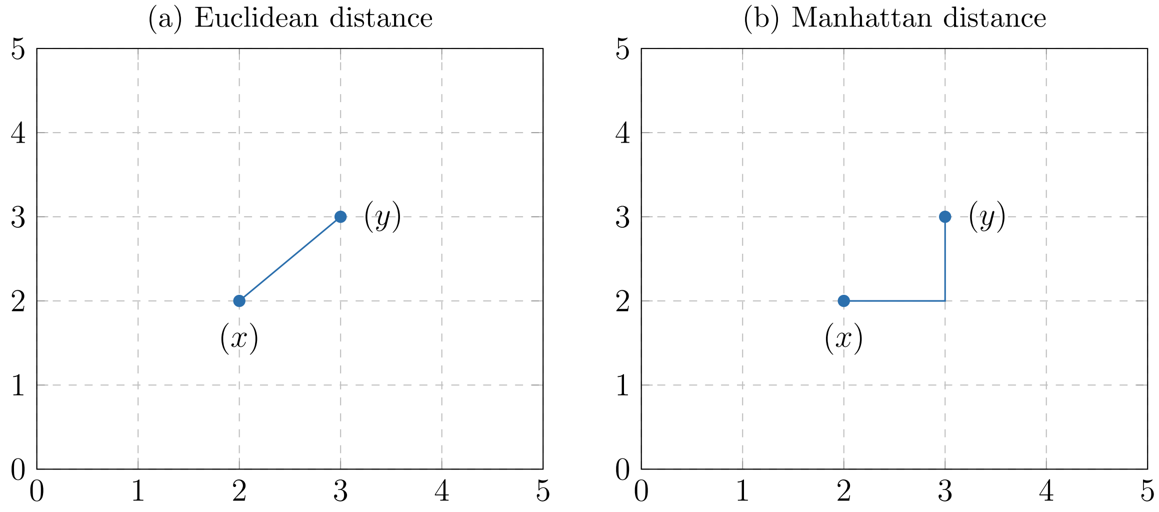


Figure 3.2: Difference between Manhattan and Euclidean distance

Another very common distance function is the *Manhattan distance* (ibid.). This distance function is closely related to the Euclidean distance, and their difference can be seen in Figure 3.2a and Figure 3.2b. The name *Manhattan distance* stems from how it relates to traveling between the tall buildings in the streets of Manhattan (another common name for this measure is the *city block distance*). The formula for the Manhattan distance can be seen in Equation 3.13, where x and y are two points in an n-dimensional space.

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{3.13}$$

The difference between the Euclidean and the Manhattan distance functions relates to how negative differences are handled. In the Manhattan distance, negative differences are directly converted to positives by taking the *absolute difference* between the values. In the Euclidean distance, negative differences are handled by squaring, which always results in a positive value. This squaring, however, introduces a side effect such that the differences are no longer in the same units as the original values. To convert the distances back to the units of the input, the square root is applied.

More generally, these distance functions can be described by the formula shown in Equation 3.14, where x and y are two points in an n-dimensional space, and r is the order of the distance function (ibid.). Using this formula, the Manhattan distance is of order *one*, whereas the Euclidean distance is of order *two*. This definition of the distance function is also known as the *L_r-norm*. An interesting property of this definition is that as $r \to \infty$, the magnitude of the largest distance gets so large that all other distances become negligible. In other words, as $r \to \infty$, the result of $d(x, y)$ becomes $max(|x_1 - y_1|, |x_2 - y_2|, ..., |x_n - y_n|)$.

$$d(x, y) = \sqrt[r]{\sum_{i=1}^{n} |x_i - y_i|^r} \tag{3.14}$$

Mean Absolute Error and Root Mean Squared Error

The *Mean Absolute Error* (MAE) and *Root Mean Squared Error* (RMSE) are two very common statistical distance metrics which are closely related to the Manhattan and Euclidean distance functions, respectively (Willmott and Matsuura 2005). The difference is mainly that instead of taking the sum of distances, the *mean* of the distances is taken. The formula for MAE and RMSE, on the form related to the formula for L_r-*norm*, is shown in Equation 3.15. The order (r) is *one* for MAE, and *two* for RMSE.

$$d(x, y) = \sqrt[r]{\frac{\sum\limits_{i=1}^{n} |x_i - y_i|^r}{n}} \tag{3.15}$$

There are, however, some semantic differences between MAE/RMSE and the L_r-*norm* distance measure. MAE and RMSE are statistical measures used to measure the mean error between the samples of a prediction model and the actual values. In other words, x and y are not points in an n-dimensional space but usually samples over a time dimension. An example of this is illustrated in Figure 3.3. Both MAE and RMSE calculates the mean of the errors (shown in red) between a prediction model and the actual values. The simplified formulas for MAE and RMSE can be seen in Equation 3.16 and Equation 3.17, respectively, where $\hat{y}$ are the predicted values of a model, y are the actual values, and n is the number of samples (ibid.). Although not used in these formulas, both the predicted ($\hat{y}$) and actual (y) values often have an associated time dimension as the x-axis.

$$MAE = \frac{\sum\limits_{i=1}^{n} |\hat{y}_i - y_i|}{n} \tag{3.16}$$

$$RMSE = \sqrt{\frac{\sum\limits_{i=1}^{n} |\hat{y}_i - y_i|^2}{n}} \tag{3.17}$$

The difference between MAE and RMSE lies in the definition of how errors (or differences) are penalized (ibid.). In MAE, errors of all sizes are weighted equally, meaning that an error of *four* is twice as bad as an error of *two*. All errors are, in other words, penalized in an additive fashion. For the RMSE metric, larger errors are penalized more than smaller ones. An error of *four* is no longer twice as bad as an error of *two*, but actually four times as bad ($\frac{4^2}{2^2} = 4$). One or a few larger errors affect the score of RMSE much more compared to MAE for the same input. Which of these metrics one should choose depends upon whether errors should be penalized in an *additive* or *exponential* fashion. If all errors should be weighted equally, then MAE is the better choice. The score of RMSE is always equal to or more than the MAE score for the same input, and never lower. The only case where they are equal is when the magnitude of all errors are equal.

Mean Percentage Error

Another common related metric is the *Mean Percentage Error* (MPE), and all of its variations (Wikipedia 2018b). The difference between MPE and, for example, MAE or

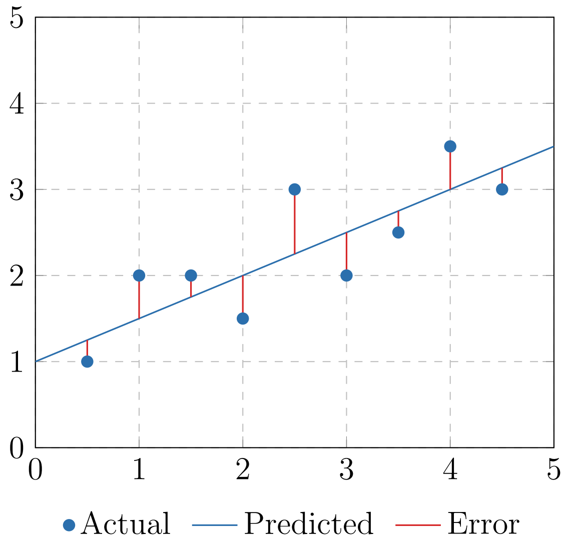


Figure 3.3: Example of forecast error for a model

RMSE is that the raw errors are converted to percentages. The formula for MPE can be seen in Equation 3.18, where $\hat{y}$ are the predicted values of a model, y are the actual values, and n is the number of samples. A variation of MPE (and more commonly used) is the *Mean Absolute Percentage Error* (MAPE) (formula shown in Equation 3.19), with the difference being that all errors are converted to positive values (Stellwagen 2011). With MPE, negative and positive errors cancel out, which is prevented in MAPE.

$$MPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\hat{y}_i - y_i}{y_i} \times 100\% \tag{3.18}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{3.19}$$

There are a few caveats worth mentioning in relation to both MPE and MAPE. The first is that the penalty for errors where the actual value is lower is higher compared to when the actual value is higher. For example, consider the predicted value *five* with the actual value *six*. The error is, in this case, *one*, and the percentage error is $16.66\% = \frac{1}{6}$. Now consider the predicted value 24 with the actual value 25. Again, the error is *one*, but the percentage error is now $4\% = \frac{1}{25}$. As seen, the penalty of the error depends upon what the actual value is. Another caveat of these metrics is that whenever the actual value is zero, the percentage error becomes *undefined*.

An alternative to MPE and MAPE is the *Weighted Absolute Percentage Error* (WAPE) (also known as the *MAD/Mean Ratio*) (Stellwagen 2011). Instead of calculating the percentage errors relative to the actual value at the given point, all percentage errors are calculated relative to the *mean* of all the actual values. This way, an error of *one* results in the same percentage error no matter what the actual value may be. The formula for WAPE can be seen in Equation 3.20, where $\hat{y}$ are the predicted values of a model, y are the actual values, $\overline{y}$ is the *mean* of the actual values, and n is the number of samples. This metric is, in fact, the same as the MAE metric divided by the mean of the actual values ($WAPE = \frac{MAE}{\overline{y}}$). *Mean Absolute Deviation*, or MAD for short, is another name for MAE, and hence why the WAPE metric is also known as the *MAD/Mean Ratio*.

$$WAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{\overline{y}}\right| \times 100\% \tag{3.20}$$

Coefficient of Determination

The *Coefficient of Determination* (also known as r^2 or R-squared) measures the proportion of variation in one variable that can be described by another variable (Minitab 2017). The *variance* of a variable is a measure of the average squared distance of the values from the variable's mean. Take a variable with the values $[1, 2, ..., 100]$ as an example. The mean of this variable is 50.5, which means that the average distance from 50.5 is 25. This example demonstrates the average *absolute difference* from the *mean*. The *variance*, however, is calculated as the *squared difference*, which might be a little harder to conceptualize, but the logic is, nonetheless, transferable. The formula for variance can be seen in Equation 3.21, where y is the variable, $\overline{y}$ is the mean, and n is the number of samples. The coefficient of determination measures, as mentioned, the proportion of variance in one variable that can be described by another variable. The squared difference between two variables is the variation in one variable that is *not* described by the other variable. In other words, this difference in relation to one of the variable's variance is the proportion of variance that is *not* described by the other variable. This means that the rest of the proportion *is* described. The formula for r^2 can be seen in Equation 3.22, where y is the first variable, $\overline{y}$ its *mean*, and $\hat{y}$ the other variable.

$$variance = \frac{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}{n} \tag{3.21}$$

$$r^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{3.22}$$

The first variable (y) is often the actual values of some measurement and $\hat{y}$ the associated predicted values. In this case, the coefficient of determination describes the accuracy of the prediction in relation to the actual values. For example, an r^2 score of 0.98 means that the prediction $\hat{y}$ has an accuracy of 98%. The name r^2 stems from the fact that, in some cases, the coefficient of determination equals the squared value of the *Pearson Correlation Coefficient* (which is described in Section 3.4.2). One example of such a case is when the prediction is derived from a least-squares regression based on the actual values. The r^2 score can, in some cases, be negative, which happens whenever the squared difference between the two variables is larger than the variance in the first variable.

3.4.2 Correlation

All the distance metrics described above require both variables in question to be in the same units of measurement. Take respiratory effort belts as an example. How tight such

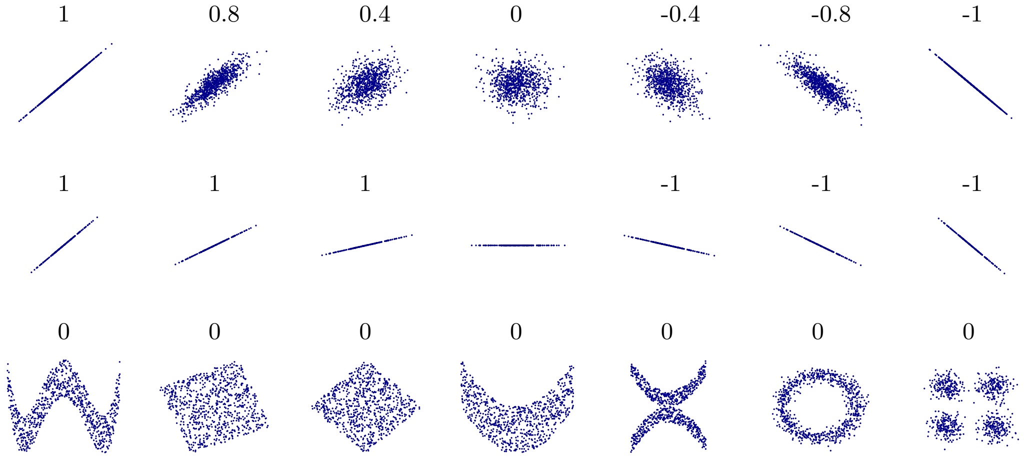


Figure 3.4: Example of how different values of Pearson's r correspond to relationships (Boigelot 2011)

a belt is fastened around a subject determines the baseline distraction value. If the belt is fastened rather tight around the subject, it may record the amplitude of breathing in the range of 60–70% distraction, and when it is fastened more loosely, it may record the amplitude of the same breathing in the range of 20–30% distraction. The breathing may be accurately captured in both cases, even though the unit of measurement is different. Directly measuring the distance between the signal from both cases is, therefore, meaningless. In cases like this, it may be more appropriate to measure accuracy based on the *relationship* between the variables. If the relationship, for example, is perfectly linear, then both variables can be converted to a common unit of measurement.

Pearson Correlation Coefficient

The *Pearson Correlation Coefficient* measures the strength of a linear relationship between two variables x and y (Kent State University 2018). If a *perfect* linear relationship exists between the two variables, then it is possible to perfectly predict one variable from the other using the linear equation $y_i = ax_i + b$, where a and b are constants. The value of this metric is often referred to as r, or *Pearson's* r, and is calculated as the covariance between two variables divided by the product of their standard deviations. The formula can be seen in Equation 3.23, where x is the first variable, y is the second variable, and $\overline{x}$ and $\overline{y}$ are their mean values, respectively. The result is a value between *minus one* and *one*, where *one* means a perfect positive linear relationship, *minus one* means a perfect negative linear relationship, and *zero* means no linear relationship. An example showing how different linear relationships are scored can be seen in Figure 3.4.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \tag{3.23}$$

The Pearson Correlation Coefficient does not assume the data to be normally distributed, but if the data is highly skewed, then the correlation coefficient will also be highly skewed.

Another factor worth considering is that the score is very much affected by outliers. One or a few large outliers can, in fact, make an otherwise obvious relationship get a very low r score, which may then be interpreted as *no relationship*.

Spearman's Rank-Order Correlation Coefficient

The *Spearman's Rank-Order Correlation Coefficient* (also known as r_s) measures the strength of a *monotonic* relationship between two variables (Spearman 1904). A monotonic relationship is perfect if one variable always increases as the other increases, or always decreases as the other increases. The degree of the increase or decrease is not constant, but the direction remains monotonic. The formula for this measure is actually the same as the formula for *Pearson's r* (shown in Equation 3.23), but with one additional step. Before the formula for *Pearson's r* is applied, all values are transformed into their corresponding *rank-values*. For example, consider a variable with the values $[8, 7, 5, 9, 6]$. The rank-order transformation of these values is $[8, 7, 5, 9, 6] => [4, 3, 1, 5, 2]$. The general formula is shown in Equation 3.24, where x and y are two variables, and n is the number of values. R_{x_i} is the rank of i-th x value, $\overline{R}_x$ is the mean of all the ranks in x, and the same applies analogously to y.

$$r_s = \frac{\sum\limits_{i=1}^{n}(R_{x_i} - \overline{R}_x)(R_{y_i} - \overline{R}_y)}{\sqrt{\sum\limits_{i=1}^{n}(R_{x_i} - \overline{R}_x)^2(R_{y_i} - \overline{R}_y)^2}} \tag{3.24}$$

In contrast to *Pearson's r*, the *Spearman's r_s* is robust against outliers. This is the case because the actual values are never used, but only their rank-order. No matter how large an outlier might be, its rank is still just one more compared the next largest value.

3.4.3 Classification

A binary classifier is a function which classifies an entity as belonging to one of two classes. For example, a classifier may be a function which determines whether or not a patient is positive or negative for a given disorder. The result of a binary classification has four outcomes: (1) a sick patient may be correctly classified as positive for the disorder (i.e., true positive), (2) a sick patient may be incorrectly classified as negative (i.e., false negative), (3) a healthy patient may be correctly classified as healthy (i.e., true negative), or (4) incorrectly classified as sick (i.e., false positive). In other words, the result is either a true or false positive, or a true or false negative. To describe the accuracy of binary classifiers, a number of metrics have been defined. These metrics are very commonly used in both medicine as well as machine learning, and four of the most common ones are defined as follows (Parikh et al. 2008):

The variables in the formulas for these metrics are given as the number of true positives as *TP*, the number of false positives as *FP*, the number of true negatives as *TN*, and the number of false negatives as *FN*.

Sensitivity

The sensitivity metric describes the proportion of correctly classified positive entities in relation to the number of entities that *should* have been classified as positive. For example, given 20 patients that are positive for a given disorder. If fifteen of these are correctly classified as positive, then the sensitivity metric for this classifier yields a score of $75\% = \frac{15}{20}$. The general formula for sensitivity is given in Equation 3.25.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.25}$$

Positive Predictive Value

The positive predictive value metric describes the proportion of correctly classified positive entities in relation to all positively classified entities. In other words, the proportion of entities classified as positive that are *true* positives. For example, given 20 true positives and five false positives, the positive predictive value for a given classifier is given as $80\% = \frac{20}{20+5}$. The general formula for positive predictive value can be seen in Equation 3.26.

$$Positive\ predictive\ value = \frac{TP}{TP + FP} \tag{3.26}$$

Specificity

The specificity metric describes the proportion of correctly classified negative entities in relation to all entities that are truly negative. For example, given 500 patients that are negative for a given disorder. If 250 of these are correctly classified as negative, then the specificity metric for a classifier yields a score of: $50\% = \frac{250}{500}$. The formula for specificity can be seen in Equation 3.27.

$$Specificity = \frac{TN}{FP + TN} \tag{3.27}$$

Negative Predictive Value

The negative predictive value metric describes the proportion of correctly classified negative entities in relation to all negatively classified entities. In other words, the proportion of entities classified as negative that are *true* negatives. For example, given 20 true negatives and five false negatives, the negative predictive value for a given classifier is given as $80\% = \frac{20}{20+5}$. The general formula for negative predictive value is shown in Equation 3.28.

$$Negative\ predictive\ value = \frac{TN}{FN + TN} \tag{3.28}$$

3.4.4 Accuracy Measures in Related Literature

Many of the metrics described above are widely used in related literature to measure the accuracy of physiological sensors. We present in this section a brief overview of how these metrics are used, and on what types of data.

Silva et al. (2015) measure the accuracy of an ECG sensor (BITalino) against a more expensive gold standard (Philips PageWriter Trim III series ECG). Put shortly, they let an expert manually annotate all the R-peaks (heartbeats) while using different algorithms to detect R-peaks automatically for the two devices. Next, they use the manually annotated R-peaks as ground truth for which they compare the automatically detected R-peaks against. To compare the accuracy of the R-peak detection of both signals, they utilize the *positive predictive value* metric (which they refer to as *precision*). In addition, they also calculate the *RMSE* and *coefficient of determination* metrics.

Retory et al. (2016) measure the accuracy of a respiratory effort belt (RIP type, NOX T3 Sleep Monitor) with the signal from a pneumotachograph as the gold standard. They extract various respiratory features from both signals, such as *tidal volume* (V_t), *inspiratory time* (T_i), and *expiratory time* (T_e), and determine the accuracy of these features based on the *Spearman's Rank-Order Correlation Coefficient* (i.e., monotonic relationship).

Seppänen et al. (2013) measure the accuracy of a respiratory effort belt (RIP type) with the signal from a spirometer as the gold standard. The metrics they utilize are the *RMSE* and *coefficient of determination.* Their goal is, however, not to determine the mere accuracy of the given sensor, but rather determine how much their noise filtering algorithm improves the accuracy. In other words, the metric's ability to yield information about how good the sensor is, is not of concern, only its comparability. For example, it is clear that an RMSE score of 22.8 is better than a score of 45.3, but not exactly *how much better* (in relation to performance), or even *how good* each sensor might be.

Liu et al. (2013) measure the accuracy of a respiratory effort belt (piezoelectric type) with a respiratory gas exchange system as the gold standard. The metrics they use include *MPE* and *RMSE.* Again, their goal is not to determine the accuracy of the given sensor, but rather determine how much their noise filtering algorithm improves the accuracy.

Cantineau et al. (1992) measure the accuracy of a respiratory effort belt (RIP type) with the integrated signal from a pneumotachograph as the gold standard. After calibrating the sensors (i.e., converting to the same unit of measure, milliliters), they calculate the accuracy of the tidal volume (V_t) of each breath by using the *MAE* and *MAPE* metrics.

Brouillette et al. (1987) compare the breath detection accuracy between the respiratory effort sensor types *RIP* and *impedance plethysmography* (IP). A reliable signal for airflow is used as the gold standard. The metrics they use to compare the accuracy of the breath detection are *sensitivity* and *positive predictive value.*

Whyte et al. (1991) measure the accuracy of respiratory effort belts (RIP type) by comparing the *tidal volume* (V_t) between the belts and the integrated signal from a pneumotachograph. The metrics they utilize include the *Pearson Correlation Coefficient.*

Adams et al. (1993) compare the breath amplitude accuracy of three different respiratory effort sensors (RIP, strain gauges, and impedance plethysmography), with the signal from an integrated pneumotachograph as the gold standard. The metric they use is the *mean* *difference* of the breath amplitudes from the sensors and the gold standard. The *mean difference* metric, also known as *Mean Bias Error* (MBE), is closely related to MAE, but negative values are not converted to positives. This means that positive and negative differences cancel out as the mean is calculated.

3.5 Respiratory Effort Sensors

For the remainder of this thesis, we focus primarily on respiratory effort sensors. In that regard, this section gives an overview of the different sensor type technologies, physiological features, and various noise filtering techniques for these kinds of sensors.

As mentioned, there are mainly two distinct types of apneic/hypopneic events, *central* and *obstructive* (a combination is also possible). In the case of a central apneic event, the patient is exerting no signs of respiratory effort at all. During an obstructive apneic/hypopneic event, however, the breathing is physically obstructed while the patient is still desperately trying to breathe (i.e., exerting respiratory effort). Respiratory effort sensors are used to detect any signs of respiratory effort such that cases of central and obstructive apneic/hypopneic events can be distinguished. The gold standard sensor for detecting respiratory effort is *Esophageal Manometry*, which is a tube inserted into the esophagus (i.e., a tube inserted into the throat stretching down to the stomach) (Berry et al. 2012). This kind of sensor is very invasive and uncomfortable for the patients, and is, therefore, rarely used in practice. Alternative sensors measuring the movement of the thorax and abdomen are mostly used instead. There are, however, only two types of respiratory effort sensors, in addition to the esophageal manometry, that are recommended by the *American Academy of Sleep Medicine* (AASM) (Berry et al. 2012). Namely *Respiratory Inductance Plethysmography* (RIP), and *Polyvinylidene Fluoride* (PVDF) type sensors.

In addition to measuring respiratory effort, these kinds of sensors can also be used as an indirect measure of tidal volume. Konno and Mead (1967) show that the respiratory process can be described as a system with two degrees of freedom (2-DOF) of motion. The sum of the movement from both the thorax and abdomen, after calibration, reflects a semiquantitative estimate of tidal volume. The calculation of RIP_{sum} is shown in Equation 3.29, where a and b are constants determined as the result of a calibration procedure. One should, however, note that even in clinical settings, calibration of the belts is rarely performed, and hence the *uncalibrated* version of RIP_{sum} is more widely used (Berry and Wagner 2014). Consequently, the constants a and b are usually both set to *0.5* so that the magnitude of the RIP_{sum} signal is comparable to the magnitude of the raw thoracic and abdominal signals. On a side note, RIP_{sum} is only used as an example, and the same description analogously applies to the PVDF counterpart PVDF_{sum} (and other counterparts).

$$RIP_{sum} = a \cdot RIP_{thorax} + b \cdot RIP_{abdomen} \tag{3.29}$$

As observed by Konno and Mead (1967), the respiratory process reduces from a system with two degrees to one degree of freedom of motion, whenever the glottis (throat) is

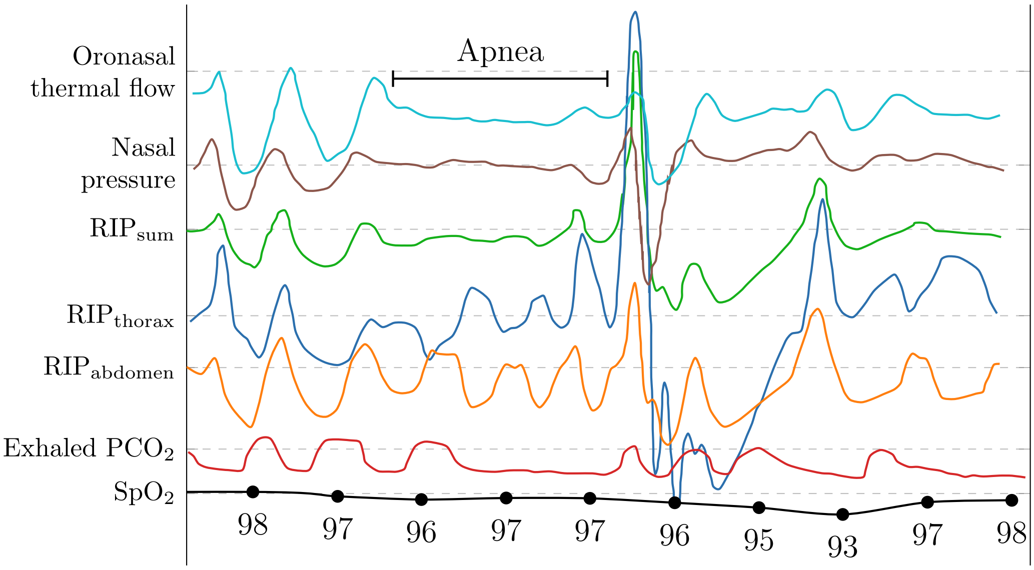


Figure 3.5: Apnea present in RIP_{sum} but not in $RIP_{abdomen}$ nor RIP_{thorax} (Berry et al. 2012)

closed. This means that an increase in volume in the thorax is equal to the opposite loss of volume in the abdomen, and vice versa. This observation is the foundation of their presented *isovolume* calibration method. Put shortly, this calibration method involves the subject holding their breath while trying to move as much air back and forth between the abdomen and thorax. The relationship between the decrease and increase of volume in the thorax and abdomen is then used to derive the a and b constants of Equation 3.29.

The respiratory effort signals captured separately from either the thorax or abdomen are not recommended by AASM (Berry et al. 2012) for apnea monitoring. The $RIP_{abdomen}$ and RIP_{thorax} (also known as *dual thoracobdominal RIP*) signals are, on the other hand, recommended for hypopnea monitoring (in addition to the RIP_{sum} signal). The main reason for this is clearly illustrated in Figure 3.5 where an apneic event is visible in the RIP_{sum} signal but neither in the $RIP_{abdomen}$ nor RIP_{thorax} signals. What happens is that the motion of the thorax and abdomen becomes asynchronous (paradoxical) during a breathing obstruction, and respiratory effort is still recorded by the belts (i.e., 2-DOF to 1-DOF). Any asynchronous behavior of the signals cancels out during the summation (i.e., calculation of RIP_{sum}), whereas any synchronous behavior (breaths) amplifies.

3.5.1 Sensor Types

There are many types of respiratory effort sensors available, and some of which include respiratory inductance plethysmography (RIP), piezoelectric belts (PZT), impedance plethysmography (IP), polyvinylidene fluoride (PVDF), and strain-gauge belts. However, only two of these are recommended for sleep apnea monitoring by the AASM (Berry et al. 2012). Namely the RIP and PVDF type belts. According to Vaughn and Clemmons (2012), the reason why piezoelectric belts are not included in these recommendations is that of the lack of formal testing, and not because of performance. For this thesis, we only have the RIP, PZT, strain-gauge belts, and IP sensor types available, and thus, solely focus on these.

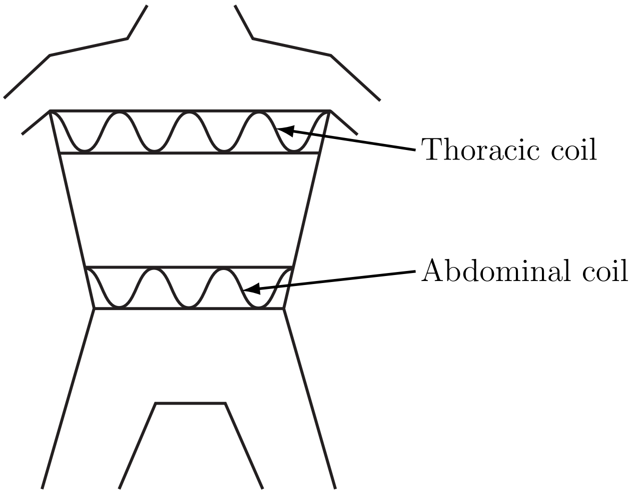


Figure 3.6: Outline of Respiratory Inductance Plethysmography (RIP) (Scilingo et al. 2011)

Respiratory Inductance Plethysmography

Respiratory Inductance Plethysmography (RIP) (Cohn et al. 1982) uses an elastic belt with an embedded coil wrapped up in the shape of a sinusoid or zig-zag pattern, stretching the whole circumference of the belt (see Figure 3.6). An alternating current is passing through this coil, which generates a magnetic inductive field, and as the belt distraction changes, so does the magnetic inductive field. The main advantage of this technology is that because the coil spans the whole circumference of the belt, it is not prone to entrapment. Even if a small part of the belt becomes trapped, the change in belt distraction is still correctly captured. The signal reflects solely the *distraction* of the belt and not the pressure/force of distraction or any other factors. Consequently, the signal reflects the change of *volume* associated with respiration.

Piezoelectric Belts

Piezoelectric belts (PZT) (Pennock 1990) consist of an elastic belt with an attached piezoelectric sensor. The sensor spans only a small area of the total belt circumference, which can be seen in Figure 3.7. The piezoelectric sensor is capturing *pressure* changes caused by changes to the distraction of the belt and does not directly reflect the circumference of the belt. The result is that the signal captures the respiratory process by changes in *airflow* rather than *volume*. The difference between volume and flow is described further in Section 3.5.2. The advantage of piezoelectric belts compared to RIP is that they are usually significantly cheaper. The main disadvantage is that the sensor part of the belt is somewhat prone to entrapment. If the patient is lying on the sensor part, for example, the belt may expand without the sensor detecting it. Until around the year 2007, piezoelectric belts were widely used in many sleep centers (Berry et al. 2012) before the RIP counterpart mostly replaced it after the recommendations from the AASM.

Impedance Plethysmography

The *Impedance Plethysmography* (IP) sensor utilizes multiple electrodes attached to the patient's thorax (Gupta 2011). A small high-frequency current is inflicted across the thorax, which changes as the thorax expands and contracts. The movement associated

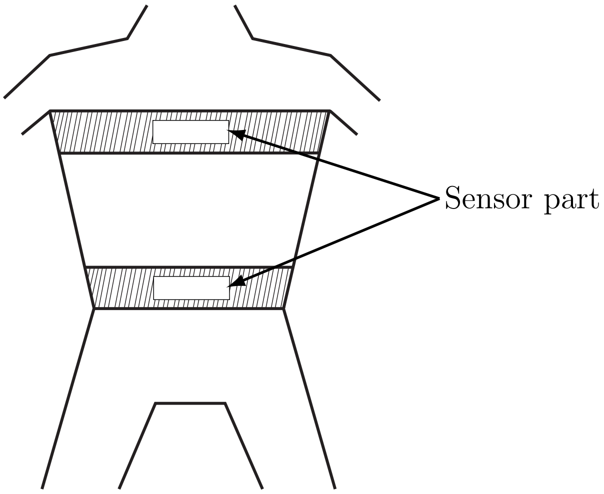


Figure 3.7: Outline of piezoelectric belts (PZT) (sensor part in white)

with breathing results in a low-frequency wave in which the higher frequency current rides on top of. The resulting signal then contains both the respiratory frequency band and the higher (inflicted) frequency band. The use of electrodes may feel slightly more uncomfortable for some people compared to the belts, but the advantage is that if the patient is already hooked up to an ECG, a few electrodes can be dedicated to capturing respiration instead. *Plethysmography* is defined as the *change of volume* (OED 2018b). Therefore, an IP sensor (like the RIP) captures the respiratory process by changes in volume.

Strain-gauge Belts

Strain-gauge belts consist of an elastic belt with an embedded conductive metal strip (Wikipedia 2018c). As the thorax and abdomen expand or contract, the metal strip becomes longer and thinner, shorter and broader, resulting in changes in electrical resistance. Because the electrical resistance reflects the current stretch of the belt, strain-gauge belts capture the respiratory process as *volume*. These kinds of belts *may* use a conductive metal strip stretching the whole circumference of the belt, but they more often span just a smaller area like PZT belts.

3.5.2 Physiological Features

Many physiological features can be extracted from the signal of respiratory effort sensors, with the main global feature being the *breath*. The shape of a breath is different depending on if the sensor is capturing airflow or volume (see Figure 3.8). Each breath in the signal also contains multiple internal features such as *total breath duration* (T_{tot}), *inspiratory time* (T_i), *expiratory time* (T_e), and *tidal volume* (V_t). As seen in Figure 3.8, tidal volume cannot be directly extracted from an airflow signal. The airflow needs to be integrated to volume before the tidal volume can be extracted. Other global features are also present in the signal such as *breath-to-breath time*, *respiration rate* (i.e., the number of breaths per minute (BPM)), and *minute ventilation* (i.e., the total volume of air inhaled and exhaled during a minute).

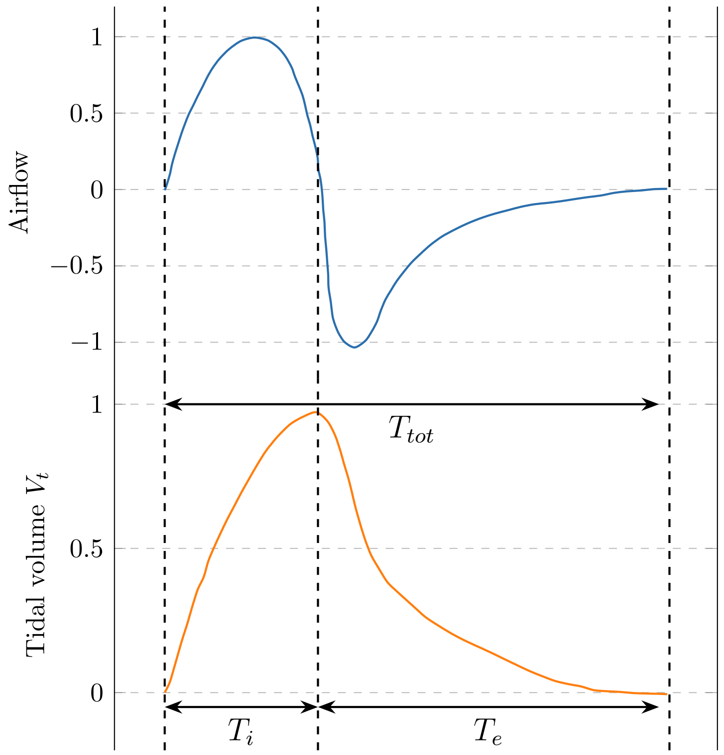


Figure 3.8: Waveform shape of airflow versus volume (McGill University 2018)

3.5.3 Signal Quality Indicators

Defining signal quality indicators for respiratory effort sensors are not trivial, and there have to the best of our knowledge not been published any either. In contrast to, for example, ECG sensors or pulse oximeters, effort sensors often only have one variable available. For example, an ECG usually consists of multiple electrodes, all capturing the same physiological process. If one of the electrodes suddenly deviates a lot from the rest of the electrodes, then it is probably faulty. A pulse oximeter has both red light, infrared light, and the pulse rate to base the quality on. If the relationship between these variables deviates from the expected norm, then there is probably a quality problem (as presented in Section 3.3).

With only one variable available, a comparison cannot be made, and the evaluation has to be done based on how the variable alone is expected to behave. A respiratory effort sensor is supposed to reflect any movement that is physically feasible. Physical movement associated with breathing and respiratory effort are expected to lie in the lower frequency bands, and any higher frequencies may, therefore, be seen as unexpected behavior. However, unless the frequencies are overlapping the breathing and respiratory effort components of the signal, then it can quite easily be filtered out and does, therefore, not impose any quality problems. Anything that can be corrected is, in other words, not of concern with respect to quality. Noise that is in the expected frequency spectrum (i.e., overlapping breathing and effort) may impose a quality problem but is also very hard to identify.

By using both an abdominal and a thoracic effort sensor, two variables capturing the same process are available. The difference between this scenario and, for example, an ECG is that different behavior is expected from time to time from the various sensors. As mentioned above, during an obstruction the breathing becomes asynchronous (paradoxical), and this kind of asynchrony should not be regarded as a quality problem.

As a result, defining *useful* signal quality indicators for respiratory effort sensors are challenging. A few indicators may, however, be defined for certain edge cases. For example, if a sensor is taking on values very close to its minimum or maximum for a duration of time, something is probably wrong. Another example may be to measure the frequency spectrum in which breaths are expected to reside. This spectrum *should* be the dominating part of the signal. Additionally, if breaths are absent from the signal for a significant amount of time, something is probably wrong.

3.5.4 Accuracy Measures

The signal from a respiratory effort sensor can be described as $X = X_R + X_N$, where X_R is the respiratory component and X_N is the noise component. The respiratory component is nominally between 0.1–0.5 Hz, but there are some respiratory components outside this range as well, for example, acute breathing or coughing (Keenan and Wilhelm 2005). The noise component can further be separated into motion artifacts, background noise, and hysteresis of the sensor, which is typically regarded as being present in the higher frequency bands (Liu et al. 2013). The magnitude of the noise component represents the accuracy of the sensor. The smaller the noise component, the more accurate the respiratory component.

The distance between two sensors capturing the same respiratory process can be described by the distance function $d(X, Y)$ shown in Equation 3.30, where X and Y are the two signals. As shown, the distance between the two signals is the difference between their noise components. If the Y signal is a *ground truth* signal, then its noise component is *zero* and its respiratory component is, therefore, 100% correct. This means that the distance between the two signals is the noise component (X_N) of X alone, and its magnitude reflects the accuracy of the signal. A ground truth is, however, not practical to acquire, and so the Y signal is instead a signal from a gold standard sensor. In this case, the assumption one has to make is that the noise component of the gold standard is as small as it can be, such that it does not affect the extraction of X_N too much.

$$d(X, Y) = X - Y => (X_R + X_N) - (Y_R + Y_N) => X_N - Y_N \tag{3.30}$$

This is only a very simple description, and it might even be too oversimplified, but it illustrates the concept. The "problem" with respiratory effort sensors is that the unit of measurement is never stable. The amplitude of the produced signal is dependent on how tight the belts are fitted around the subject. Even after a calibration procedure, the amplitude continues to vary in relation to movement, often due to belt slippage, body position changes, etc. (Whyte et al. 1991). Consequently, the process of isolating the noise component is much more convoluted in practice compared to the example above. As such, the more practical way of measuring distance between respiratory effort sensors may be to measure the distance of various physiological features (as presented in Section 3.5.2) rather than the raw signals directly.

3.5.5 Noise Filtering

The primary quality problem with respiratory effort sensors is the presence of *motion artifacts*. The sensors are designed to capture the motion associated with breathing and respiratory effort, but the side effect is that they capture any kind of motion. Measuring respiration with a RIP belt while a person is running, for example, is almost impossible to do. The traditional way of filtering noise from a respiratory effort signal is with the use of a low-pass filter. With the respiratory component residing nominally between 0.1–0.5 Hz, a cut-off frequency of 1–1.5 Hz is typically used (Keenan and Wilhelm 2005). The main drawback with a low-pass filter is that it may also attenuate higher frequency respiratory components, such as coughing or acute breathing. Many alternative filtering methods have been proposed to improve the performance of respiratory effort sensors during mild physical activity. For example, *wavelet decomposition* (Keenan and Wilhelm 2005), *adaptive filtering* (Keenan and Wilhelm 2005), *noise discrimination* (Retory et al. 2016), and *empirical mode decomposition* (Liu et al. 2013). During sleep, however, physical movement is not too much of concern. The signal may become corrupted by noise while the patient is changing sleeping position, but the majority of the signal remains unaffected. In the context of a traditional polysomnography sleep study, the recommended cut-off frequencies for respiratory signals are 0.1 Hz for high-pass filters and 15 Hz for low-pass filters (Tripathi 2008).

3.6 Discussion and Conclusions

Based on the background material presented in this chapter, we draw the following conclusions:

- There are three means of measuring the quality of physiological time series data, namely by *completeness*, *validity*, and *accuracy*. These dimensions are summarized as follows:
  - **Completeness**
    The completeness is the proportion of the actual length of a signal in relation to its intended length. Completeness can, for example, be affected by packet loss because of wireless transmission.
  - **Validity**
    The validity of a signal represents the proportion of time the signal is behaving as expected. Validity is often threshold based, and whenever the signal takes on unexpected values or behavior that exceeds this threshold, then it is deemed invalid. As a result, validity can often be a purely intra-signal measure, which makes it suitable for real-time quality feedback/indications.
  - **Accuracy**
    The accuracy of a signal is measured as the *distance* between it, and a ground truth or gold standard signal. How distance is measured depends on the type of data in question, which can, for example, be the raw signals or any extracted features.
- It is challenging to define useful signal quality indicators for respiratory effort sensors. As a result, the validity dimension may be excluded, which means that the

completeness and accuracy dimensions are the only remaining dimensions for these kinds of data.

- All the accuracy metrics presented in Section 3.4 are, in fact, ratio scaled metrics. The reason is that they all have a clear definition of a zero point. A distance of *zero* can be interpreted as *no error*. Moreover, a perfect relationship (correlation) and a 100% accurate classifier, also analogously mean *no error*. Hence, ratio scaled.

- The best choice of accuracy metrics depends upon the nature of the data in question. For example, a sensor's ability to detect breaths can be regarded as a classification, whereas a raw signal comparison is better measured as a distance.

Chapter 4

Requirement Analysis

The first goal of this thesis is to determine how the signal quality of respiratory effort sensors can be measured in relation to sleep apnea monitoring. Once established, we assess the signal quality of four different respiratory effort sensors through a quantitative study with data from various external subjects. In this chapter, we explore what a good signal quality means in the context of apnea monitoring, along with a requirement analysis of the experiment and a description of the different sensor platforms.

We begin in Section 4.1 by exploring what the notion of a good quality signal from respiratory effort sensors is in relation to sleep apnea monitoring. We continue in Section 4.2 by defining which requirements need to be fulfilled by a peak in the signal to be regarded as a breath for the automatic breath detection algorithm, followed by requirements for the experiment (e.g., setting, number of subjects, etc.) in Section 4.3. Next, we describe the sensor platforms we assess in more detail in Section 4.4, before we summarize and conclude the chapter in Section 4.5.

4.1 Sleep Apnea

When the signal quality of a sensor is to be determined, it is important to properly define what a good signal quality represents. In the case of respiratory effort sensors, the notion of what a good quality signal is varies significantly between different use cases. For example, if the intention is to thoroughly measure the pulmonary function of a patient, then an exact waveform is a requirement to accurately estimate respiratory parameters such as inspiratory time (T_i), expiratory time (T_e) and tidal volume (V_t). On the other hand, if the goal is to solely estimate the respiration rate of a patient, neither an exact waveform nor any of these parameters, are very important factors to consider. Thus, it is essential that we analyze the sensor requirements in terms of sleep apnea monitoring before we decide on how the signal quality of the sensors should be measured.

4.1.1 Apnea Detection

In the case of apnea detection (complete cessation of breath), the feature of interest is solely the presence of gaps between two consecutive breaths, lasting a minimum of ten seconds. Thus, the waveform shape and other parameters are, in fact, irrelevant in the case of apnea detection. Given, for example, a respiration belt with a very non-deterministic

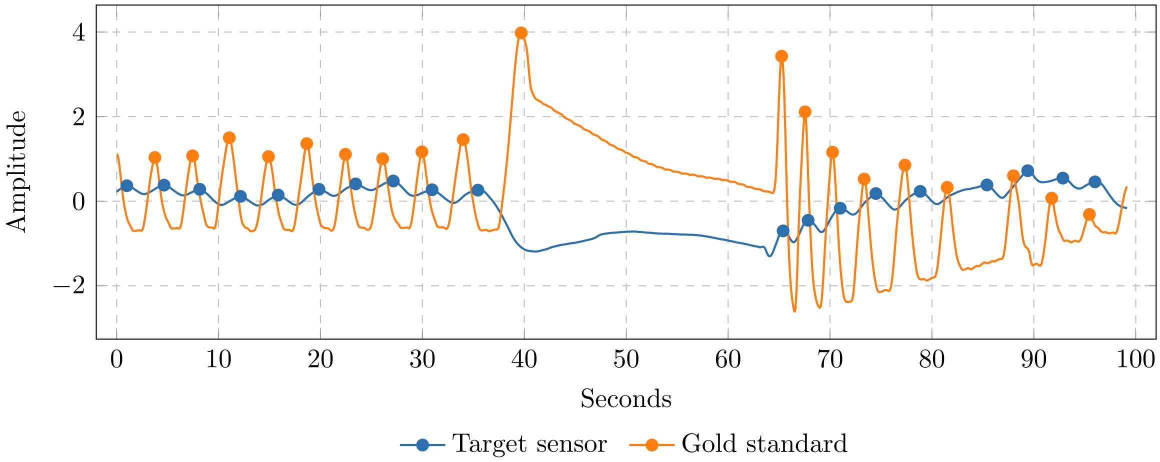


Figure 4.1: Different waveforms — gap detection still accurate

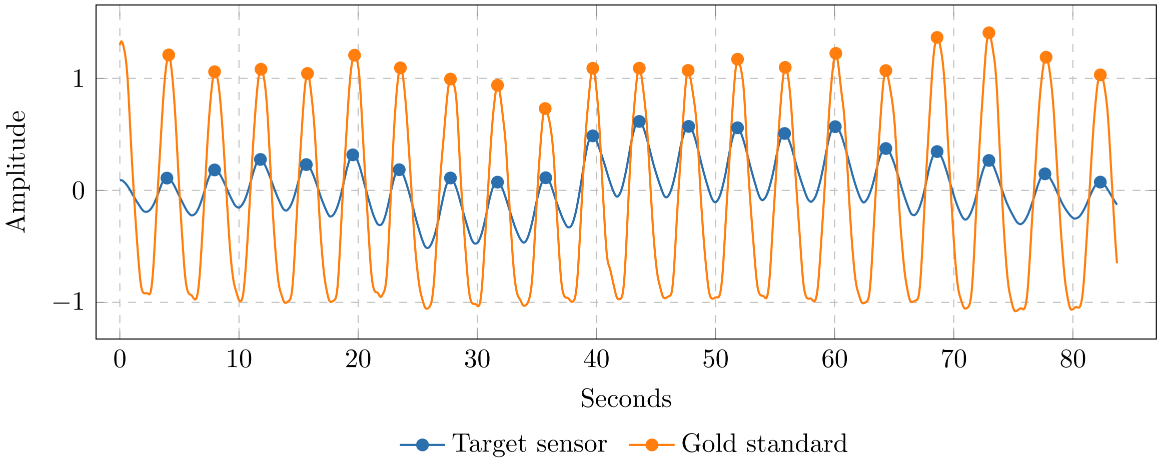


Figure 4.2: Accurate breath detection — different waveforms

and arbitrary waveform (e.g., a lot of low-frequency noise and breath amplitude variations). Even if this sensor has a very inaccurate waveform and looks very different from a gold standard, it could still be considered a good quality sensor for apnea detection. This is, however, only as long as it is able to accurately make out every gap in the signal (see Figure 4.1 and Figure 4.2). Although in reality, inaccurate breath detection is very likely to influence the accuracy of gap detection, and as such, breath detection may be the preferred metric after all. The main reason is that the noise component is generally independent of the respiratory component of the signal, and is, thus, likely to be somewhat evenly distributed.

There is, in other words, a hierarchy of quality metrics, with the accuracy of gap detection being the most fundamental. An exact waveform does indeed equal a perfectly accurate detection of both breaths and gaps. However, a perfect breath detection is not intrinsic for a perfect gap detection, nor is an exact waveform intrinsic for a perfect breath detection. What this means is that if a sensor is able to perfectly detect gaps, neither improving its waveform nor breath detection, will make it perform any better at detecting apneas.

The kinds of errors likely to be experienced in the case of apnea detection are a number of false breaths, missed breaths, or both. The breaths at the start and end of a gap can also be slightly delayed, making the gap appear shorter or longer than it really is. The signal quality of a sensor, therefore, needs to be determined relative to the tolerance for these kinds of errors. For example, an adult's respiration rate (*RR*) is typically in the range of 12–18 breaths per minute at rest. A breath lasting as long as five seconds is, therefore, not uncommon. This means that in the worst case, a false breath can occur in the middle of a 20-second gap, dividing it into two gaps of 7.5 seconds each. With a stable error rate of one false breath per minute, the sensor would in the worst case, therefore, only accurately detect gaps of 25 seconds or longer (although, not as one consecutive gap but multiple smaller ones).

So how should the accuracy of a sensor's ability to accurately detect breaths be determined? The naïve way is to directly compare the *respiration rate* (RR), which is measured as the number of *breaths per minute* (*BPM*), with the gold standard. By comparing the number of breaths per minute, it becomes trivial to reason about the interpretation of the error rate of a sensor (e.g., the error rate is 1–2 breaths per minute). The drawback of comparing the respiration rate is that missed breaths and false breaths cancel out. For example, given five false breaths and six missed breaths, the error rate is still only one breath per minute. A solution is to match the breaths of a target sensor with the breaths of a gold standard and then count the number of false positives/negatives. To determine which breath from one signal corresponds to which breath from the other, we can define that a breath is matched if its peak is between the start and end of a breath in the other signal.

In a related study by Brouillette et al. (1987), they propose the use of *sensitivity* and *positive predictive value* (see Section 3.4.3). The *sensitivity* gives the proportion of correctly identified breaths in relation to the total number of real breaths (as detected by the gold standard). In other words, if a sensor does not miss any real breaths, then the *sensitivity* would be 100%. The *positive predictive value* gives the proportion of correctly identified breaths in relation to all detected breaths. To give an example of these, let there be an extra false breath for each real breath detected by the target sensor. This would yield a *sensitivity* of *100%* and a *positive predictive value* of *50%*. In regards to sleep apnea, a low *sensitivity* and a high *positive predictive value* would indicate that *false positive* apneic events are likely to occur (i.e., false gaps). Conversely, a high *sensitivity* and low *positive predictive value* would indicate that *false negatives* are likely. An alternative or additional approach to these can, for example, be to calculate the proportion of "*clean*" minutes in the signal. If it is a known fact that any minute that contains a false breath cannot be trusted, any other minute would be *clean*, and the proportion of these minutes would be the proportion of the signal that can be trusted. Let, for example, any minute that contains one or more false breaths to be regarded as "*dirty*." Assuming that 90% of the minutes in the signal are deemed "*clean*," this means that at least 90% of the signal can be trusted in regards to apnea detection.

4.1.2 Hypopnea Detection

In the case of hypopneas, the feature of interest is a 30% (or more) reduction in airflow, lasting a minimum of ten seconds. To be able to detect reductions of airflow accurately, the breath amplitudes as recorded by a sensor need to be linear in relation to the thoracic or abdominal expansion. In other words, the breath amplitudes are linear in relation to belt distraction for a good quality signal. High amplitude variations and different types of relationships (such as monotonic or non-linear) affect the accuracy negatively. Accurate breath detection is, however, a prerequisite before it is even reasonable to start measuring the breath amplitude linearity.

When the breath amplitudes are linear in relation to the belt distraction, then a 30% lower amplitude corresponds directly to a 30% reduction in airflow. If the relationship is monotonic, however, a 30% lower amplitude could correspond to only a 15% reduction in airflow. Even worse, if the relationship is non-linear, a 30% lower amplitude could correspond to a 30% *increase* in airflow. Amplitude variations are also very important to consider. Even if the relationship is linear, a high variation lowers the accuracy of hypopnea detection considerably.

The measure of breath amplitude linearity should preferably yield an error rate as a percentage, as that is how hypopneas are scored. This would result in a metric that is rather trivial to interpret, e.g., an error rate of 5% means that the reported reduction in airflow is within 5% of the actual reduction in airflow. In other words, a measure of *variation* in a linear relationship. The way this can be done is to fit a linear regression model to the breath amplitudes of the target sensor and the gold standard. The error is then calculated as the distance from a point in the target sensor to the regression line. The next issue is how to summarize these errors into an error rate which represents the whole signal. In other words, should one take the *mean*, *min/max*, or some other measure? The drawback of taking the min/max is that outliers profoundly influence the result. In a study by Cantineau et al. (1992), they instead propose the use of *accuracy* and *precision*, which is the *mean difference* and *standard deviation*, respectively.

4.1.3 Requirements

Based on the analysis above, we have identified the following requirements:

- The signal quality in relation to apneas should be based on *breath detection accuracy.*
- The signal quality in relation to hypopneas should be based on *breath detection accuracy* and *breath amplitude accuracy.*
- The interpretation of the metrics should be trivial in relation to the sensor's ability to detect apneas/hypopneas.

Even though the gap detection accuracy of a sensor is the most fundamental quality parameter for apnea detection, it is irrelevant for the detection of hypopneas. Given that OSA sufferers usually experience both apneic and hypopneic events, we focus on breath detection accuracy instead. We illustrate what we mean by *"The interpretation of the metrics should be trivial in relation to the sensor's ability to detect apneas/hypopneas"* with an example. Given a metric about the strength of a linear relationship which yields a score of *0–1* where *0* means *no relationship* and *1* means *perfect relationship.* If a sensor gets a score of *0.86*, one has simply no means of knowing how good it is at detecting hypopneas. All one can say is that it is probably better than a sensor with a score of for example *0.67*. To know how much better it is though, the measurement scale (i.e., ordinal, interval, ratio) must also be known. If one, on the other hand, has a breath amplitude error of 5%, one knows that the sensor can at least correctly identify hypopneas with a 35% decrease in airflow (i.e., real airflow reduction is 35%, but the sensor detects a value in the range of 30%–40%).

4.2 Breath Detection

To speed up the process of scoring and extracting breaths from the recorded signals, we implement an automatic breath detection script used to aid the manual process. In this section, we define specific requirements for this breath detection script. Breath detection in a respiratory signal is by definition the same as peak detection, constrained by physical limitations. A peak is commonly defined as a value or period of a time series that is higher than its immediate neighbors. How much higher it has to be is, on the other hand, often user defined in relation to the given context. Whereas a peak can be as short as two milliseconds in duration, a breath cannot. Thus, we have to assert that the duration of each peak is within the minimum and maximum duration of a breath. What the minimum and maximum duration of a breath are, depends upon the range of respiration rates we want to support. For example, with a range of 5–100 breaths per minute, the minimum breath duration is 0.6 seconds, and the maximum is twelve seconds. Therefore, we need to define a duration interval in which each breath should be, to more accurately distinguish real breaths from noise and motion artifacts.

For a peak to be regarded as a breath, we define the following requirements:

- It must last between 0.6 and twelve seconds (duration).
- The amplitude of the peak must be at least 10% of the mean breath amplitude.
- Breaths cannot overlap.

Although a respiration rate of 100 breaths per minute is unlikely during regular sleep, such a breathing rate can indeed occur for shorter periods of time (e.g., a few seconds). An increased respiration rate is, in fact, very common after apneic/hypopneic events as the subject may be gasping for air. Likewise, longer breaths can happen as a result of, for example, an obstruction during exhalation. Moreover, some additional padding can also be useful depending on whether we want to prioritize detecting real breaths or avoiding false breaths. The threshold of 10% is set based on the scoring rules by the AASM (Berry et al. 2012). A period is considered an apnea (rather than a hypopnea) if the signal excursion (breath amplitude) is below 10% of the baseline throughout the event. The amplitude of a generic peak is commonly measured based on its y-value. However, if the signal is suffering from baseline wander, the peak's y-value may just as well be negative. The amplitude of a breath should, therefore, be measured as the difference between the peak and the start/end of the breath. We elaborate further on the details of this in Section 5.1.2. There are also additional requirements that can be set, for example, inspiration and expiration time. In other words, there are physical limitations to how fast an inspiration/expiration can happen. However, in the presence of noise, the peak of a breath can be shifted, which can make a real breath (as detected by a sensor) violate such requirements. Automatic breath detection is, in other words, not trivial to do with noisy data.

4.3 Experiment Design

4.3.1 Privacy Declaration

Whenever external subjects are to be involved in an experiment, a concern for privacy arises. The data we collect from the subjects in this study are *gender, age, weight, height*, and the *respiratory data* from the sensors. None of these attributes are regarded as *sensitive* because a single person cannot be identified based on these attributes alone, which means that they are not covered by the Norwegian law, *personopplysningsloven* (Lovdata 2000). Regardless, we require all participants to sign a written consent, accepting that we can use the data in this work, as well as in potential future work. All the gathered data are also pseudo-anonymized and stored securely according to the guidelines by Datatilsynet (2015).

4.3.2 Representativeness

Signal capture sessions performed overnight while the subjects are asleep, including both healthy as well as sleep apnea patients, are with no doubt the most representative sessions for sleep apnea monitoring. The main reason is that there are certain events and characteristics of these kinds of sessions that are very likely to influence the results of the signal quality evaluation. For example, given a sensor which simply cannot flatline. In other words, when a subject stops breathing, the sensor starts to act weirdly and produces events that *can* be regarded as breaths. If breathing stops are not included in the signal capture procedure, such an issue will not be discovered. These longer overnight sessions, however, require significantly more work, both with respect to execution as well as to subject recruitment, compared to shorter sessions that can be performed in a laboratory during wakefulness. A decent quantity of signal captures is also essential to be able to generalize about the results (i.e., such that the results represent the majority of cases). As such, shorter sessions that can be performed in a laboratory during wakefulness are preferred.

As previously stated, apneic events are described as periods with *no breathing*, lasting a minimum of ten seconds, whereas hypopneic events are defined as periods with *shallow breathing*, lasting a minimum of ten seconds. Additionally, periods of deep breathing are something that commonly happens after an apneic/hypopneic event, as the subject may be gasping for air. To simulate overnight sessions during wakefulness in a laboratory, all of these events should be included in the signal capture procedure. Apart from the characteristics directly related to sleep apnea, the subject would also be lying in bed and possibly changing sleeping position throughout the night. This behavior might just as well affect the signal from the sensors, and hence should be included in the signal capture procedure as well.

To make these shorter sessions as representative as possible with respect to sleep apnea monitoring, we define the following requirements:

- A signal capture must include:
  - Breathing stops.
  - A period of shallow breathing.
  - A period of deep breathing.
  - Multiple sleeping positions.
- The subject must be lying in bed.

In the case of traditional polysomnography, the subjects are often very restricted regarding movement and position changes. However, position changes can indeed occur with the use of less restrictive equipment and are, thus, a requirement for our signal capture procedure. For more details regarding the relevancy of different body positions, see Section 5.3.1. Another factor that might be obvious is the duration of the captures. If a sensor's signal quality degrades over time, the results of shorter sessions are indeed biased. The choice of signal capture duration is discussed further in Section 5.3.2, with the results of the preliminary testing in mind (Section 5.2).

4.3.3 Quantitative Study

For the experiment results to be regarded as useful, they must be generalizable. In other words, the results must be valid for and represent the majority of cases. To be able to do so, the sample size (number of subjects and signal captures) must be big enough to capture the general case. Additionally, variation amongst the subjects is also vital, such as gender, age, height, and weight. The required sample size varies significantly between different situations and contexts. More subjects are regarded as better, however, too many become infeasible in our limited time frame.

To better be able to reason about the minimum number of subjects we should include in our experiment, we study how many subjects are included in related work. In some related work, the number of subjects involved is often somewhere in the range 3–13, however, with some deviations. See Table 4.1 for an overview of the number of subjects used in some of the related work. Based on this information, a number of *5–10* subjects is a sufficient *minimum* requirement for our experiment.

4.4 Platforms

We have a total of five different platforms available for this thesis: BITalino, Shimmer, RespiBAN, FLOW, and NOX T3. The former four are our target sensors which we measure the signal quality of, whereas the NOX T3 is our gold standard. In this section, we give a brief description of all of these platforms, including their available sensors.

Study	**# of subjects**
(Katz and Dinner 1992)	3
(Pennock 1990)	4
(Whyte et al. 1991)	8
(Wu et al. 2009)	10
(Seppänen et al. 2013)	10
(George et al. 1988)	11
(Kogan et al. 2016)	12
(Cantineau et al. 1992)	13
(Adams et al. 1993)	20
(Brouillette et al. 1987)	28
(Retory et al. 2016)	30
(Liu et al. 2013)	105

Table 4.1: Overview of the number of subjects included in related studies

4.4.1 BITalino

BITalino is a product and an open source platform designed for people who wish to learn and prototype their own wearables (BITalino 2018a). They currently supply three different kits for the BITalino platform, each with various form factors and included sensors. There is a wide range of sensors included in these kits, such as ECG, EEG, EMG, EDA, and an accelerometer. Additional sensors can also be purchased separately, for instance, pulse oximeters and respiratory effort belts (both PZT and RIP type belts). The kits are currently priced between *150–200 Euros*, and BITalino's vision is to make BIO-signals available to anyone at a low cost. The BITalino device itself has six analog input channels available, which means that six different sensors can be used at once per device. Four of these inputs have a *10bit* resolution, while two of the inputs have a *6bit* resolution. Each of these inputs supports a sampling rate of *1 Hz, 10 Hz, 100 Hz*, or *1000 Hz*. The device is battery-powered and transmits data wirelessly over Bluetooth, but it does not have any on-board storage. In other words, the device is portable but requires an additional device such as a smartphone or a computer to persist the data. We must emphasize that the BITalino platform is not certified as medical grade equipment, nor designed to be used for medical diagnosis or in a medical setting. It is intended to be affordable to make BIO-signals available to anyone.

In our experiments, we use a *BITalino Plugged Kit BLE* (BITalino 2018e) with *piezoelectric* type respiratory effort belts (PZT) (BITalino 2018b). These belts are affordable with a price of about *95 Euros* per belt, making them suited for personal consumers. As described in Section 3.5.1, a piezoelectric respiratory effort belt captures the inhaled and exhaled *airflow* through the change of *force* (due to stretching of the belt) around the abdomen and thorax. The sensor part of these belts span only a small area of the belts, and hence the force captured by the sensor may or may not reflect the actual change of circumference around the subject. The sensor part of the belts may become trapped (e.g., by lying on the sensor), while the non-sensor part stretches independently.

RIP type respiratory effort belts do not have this issue, as the sensor part of these belts span the whole circumference. RIP type belts are, however, significantly more expensive compared to their PZT counterparts, with the RIP belt available for BITalino priced at about *750 Euros* (BITalino 2018c). Consequently, RIP type belts may be less suited for personal consumers.

The BITalino platform is intended for developers and people who want to make their own wearables. As such, data acquisition is usually done through a custom-made application with its provided *software development kit* (SDK). BITalino provides SDKs for a wide range of programming languages and platforms, such as Android, iOS, Java, C#, Python, and MATLAB. Additionally, BITalino also provides a ready-made acquisition software called *OpenSignals*, which is available for Windows, Mac, Linux, and Android. However, the Android version does currently only support real-time visualization, and the acquired data cannot be stored for later retrieval. Moreover, the desktop versions of the OpenSignals software require a custom Bluetooth dongle to work with the BITalino kits equipped with a *Bluetooth Low Energy* (BLE) type module, which is the type we are using. Fortunately, a custom-made Android acquisition application has already been created by Gjøby (2016). This application supports both network streaming of the acquired data as well as storing it locally to a file. We, therefore, use this application along with an Android device (Google Pixel) to record data from the BITalino sensors in our experiments.

4.4.2 Shimmer

Shimmer is an open source wearable sensor platform which includes sensors such as ECG, EMG, EDA, accelerometer, gyroscope, and altimeter (Shimmer 2018a). The device is battery-powered and can either transmit data wirelessly over Bluetooth or store it locally to its on-board storage. The Shimmer engineering team are highly focused on quality of the sensors, which results in a higher quality product, but consequently also a higher price. The Shimmer ECG unit costs about *500 Euros*, making it slightly more expensive than the BITalino platform. In contrast to BITalino, Shimmer is, in fact, certified as medical grade equipment due to its quality focus (after the ISO 9001 and ISO 13485 standards) (Shimmer 2018a).

Shimmer does currently not offer respiratory effort belts, but respiratory effort is captured using an ECG unit. In other words, the sensor capturing respiratory effort is of type *impedance plethysmography* (IP) (see Section 3.5.1), which means that it uses ECG leads (electrodes) attached directly to the subject's skin. Electrodes may feel a little uncomfortable for some people and can be a little harder to fit correctly without guidance from trained personnel. Additionally, the IP type sensor is regarded as being more affected by noise, baseline wander, and motion artifacts; at least compared to RIP type belts (Brouillette et al. 1987).

As Shimmer is an open platform, various SDKs are provided for a wide range of programming languages and platforms, for example, LabView, MATLAB, and Python. In addition, Shimmer also provides a ready-made data acquisition software called *Consensys*, which is available for Windows. Consensys can be used to configure the Shimmer device, stream data, and visualize the data in real-time. After the device is correctly configured, it can also record data directly to its on-board storage, making it independent of other equipment (smartphone, PC, etc.). For our experiments, we use the Consensys application to configure and manage the device, while we record the data to the device's on-board storage.

4.4.3 NOX T3

NOX T3 is a complete medical grade portable respiratory sleep monitor made by NOX Medical (2018). It sports a slim form factor, battery-powered device, with on-board internal storage. The sensors supported by NOX T3 include dual thoracoabdominal respiratory effort belts (both RIP and PZT types), ECG, nasal pressure, pulse oximeter, accelerometer, snore sensor, and more. NOX T3 is widely used in hospitals and sleep centers for the diagnosis of sleep-related disorders around the world. NOX T3 is priced at around *55 000 NOK*, making it too expensive for personal consumers. It is our gold standard device in which we compare the other sensors against for the experiments we are conducting.

NOX T3 is a proprietary platform, which means that only the software supplied by NOX Medical themselves are supported, and no SDKs are available. The provided software is called *Noxturnal*, which is only available for Windows (NOX Medical 2018b). Noxturnal can be used to configure the device and download and analyze the recordings. The recordings can also both be annotated, scored, and organized/stored from within the software itself. We, however, use it only to export the *raw* data so that we can compare it to the other sensors. A limitation of the Noxturnal software is that only recordings that are longer than about *seven minutes* are supported. Shorter recordings cannot be downloaded, and are, thus, *lost.*

For our experiments, the only sensors we use with the NOX T3 are RIP type dual thoracobdominal respiratory effort belts.

4.4.4 RespiBAN

RespiBAN is a configuration of the biosignalsplux platform, and this platform is designed for researchers to collect and analyze reliable high-definition BIO-signals (biosignalsplux 2018a). BITalino and biosignalsplux are, in fact, two different platforms made by the same company (PLUX 2018). As a result, most of the available sensors for both platforms are the same, their SDK interface is the same, and both platforms support the OpenSignals software. While BITalino is designed as a *do-it-yourself* platform, biosignalsplux is designed as a *ready-to-use* platform with a high focus on quality, which consequently also results in a significantly higher price. In contrast to BITalino, the sampling rate of the biosignalsplux device can be freely set to anything up to 4000 Hz, and the analog signal resolution is 16bit. It is also battery-powered and transmits the recorded data over Bluetooth. Due to the quality focus, biosignalsplux is, in fact, certified as medical grade equipment (after ISO 13485) (BITalino 2018d). However, the producer gives the following disclaimer concerning both BITalino and biosignalsplux: *PLUX's products are intended for use in life science education and research applications; they are not medical devices nor are they intended for medical diagnosis.*

The biosignalsplux configuration we use in our experiments is a *RespiBAN Researcher* (biosignalsplux 2018b). This particular configuration is priced at about *1 250 Euros.* Moreover, it is specifically tailored for respiratory monitoring and is, therefore, a little more restricted compared to the generic configurations. The sensors are hardwired to the device, and it includes a single RIP respiratory effort belt and a triaxial accelerometer. The maximum sampling rate is limited to 1000 Hz (instead of 4000 Hz), but the analog resolution remains at 16bit. Like with BITalino, we record the data from this device using the Android application by Gjøby (2016).

4.4.5 FLOW

The FLOW sensor is a newly developed affordable respiratory effort belt from SweetZpot (SweetZpot 2018), priced at about *200 Euros.* The sensor captures both respiratory effort using strain-gauges and also the heart rate when the belt is worn directly on the skin. It is not intended to be used as a medical diagnostic tool, but rather during physical activity such as cycling, rowing, singing, and the like. The device is battery-powered and transmits the recorded data over Bluetooth to a smartphone. For our experiments, we use the supplied Android application called *RawDataMonitor* which captures the raw respiratory data with a sampling rate of 10 Hz.

4.5 Discussion and Conclusions

To summarize, we have a total of four different types of respiratory effort sensor at our disposal: RIP, PZT, strain-gauge, and IP type sensors. Our gold standard, NOX T3, uses *RIP* type dual thoracoabdominal belts, BITalino uses PZT type dual thoracoabdominal belts, RespiBAN uses a single RIP belt, FLOW uses a single strain-gauge belt, and Shimmer uses an IP type sensor. Of the target sensors, BITalino and FLOW are the most affordable platforms, whereas Shimmer and RespiBAN are more expensive and certified as medical grade equipment.

Regarding the signal quality metrics and experiment design, we have identified the following requirements:

- Metric requirements:
  - The signal quality in relation to apneas should be based on *breath detection accuracy.*
  - The signal quality in relation to hypopneas should be based on *breath detection accuracy* and *breath amplitude accuracy.*
  - The interpretation of the metrics should be trivial in relation to the sensor's ability to detect apneas/hypopneas.
- For a peak to be regarded as a breath for the automatic algorithm, we define the following requirements:
  - It should last between 0.6 and twelve seconds (duration).
  - The amplitude of the peak should be at least 10% of the mean breath amplitude.

  - Breaths cannot overlap.

- Experiment design requirements:
  - A signal capture must include:
    - Breathing stops.
    - A period of shallow breathing.
    - A period of deep breathing.
    - Multiple sleeping positions.
  - The subject must be lying in bed.
  - A minimum of 5–10 subjects should be included.

Chapter 5

Design

This chapter presents the design of the signal quality measurement process for respiratory effort sensors. This design involves two main parts: (1) the design of the signal quality measurement process itself, and (2) the design of the script for the signal capture procedure. The design of the signal quality measurement process involves the choice of comparison methods and parameters (i.e., metrics) based on the requirements from Chapter 4, as well as any signal preprocessing steps that may be needed. The design of the signal capture procedure script defines how the data from the sensors are acquired. This includes, for example, how long each signal capture should be, what positions the subject should undertake, and what actions the subject may perform throughout the procedure. The goal is, in other words, to gather as many representative captures as necessary to make the results as reliable as possible.

We begin in Section 5.1 by defining the design of the signal quality measurement process. This includes any preprocessing steps of the raw signals, feature extraction, and the quality metrics themselves. Next, we describe a number of preliminary tests along with their results in Section 5.2, before we define the script for the signal capture procedure in Section 5.3. The design of this script is based on the results of the preliminary tests, and the choice of setting and number of subjects are based on the requirements from Chapter 4, and also anchored in related studies. Finally, we summarize and conclude the chapter in Section 5.4.

5.1 Signal Quality Measurement

5.1.1 Preprocessing

Before the signal quality metrics can be calculated, certain preprocessing steps of the raw signals must be performed. This includes, for example, synchronization, resampling, and standardization/normalization. In this section, we elaborate on the details of these signal preprocessing steps.

The RIP$_{\text{sum}}$ Signal

As described in Section 3.5, any behavior that is synchronously captured by both the thoracic and abdominal signals is amplified during the calculation of the RIP$_{\text{sum}}$ signal, while anything else either cancels out or diminishes. In other words, the amplitude

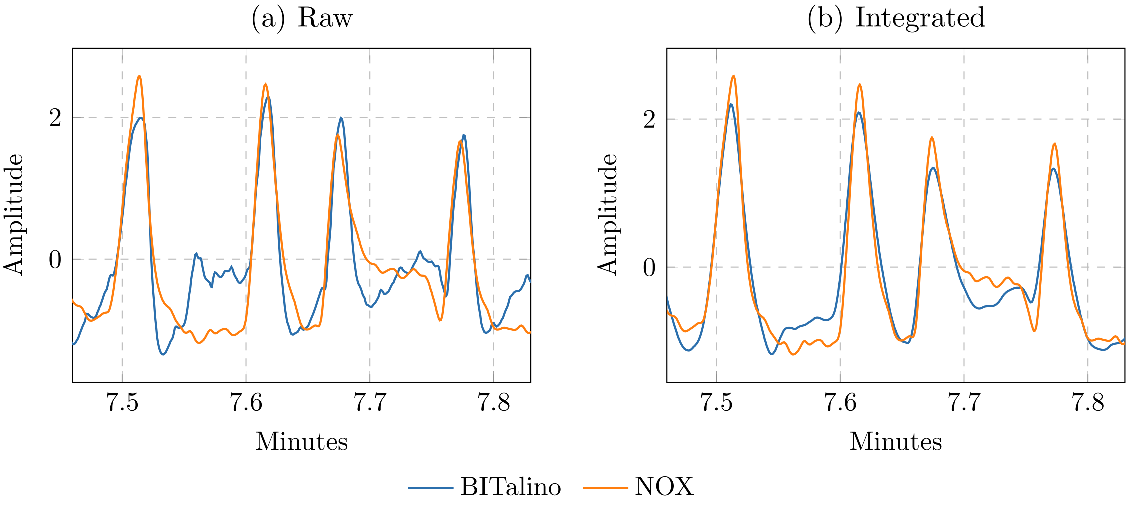


Figure 5.1: Raw versus integrated signal

of breaths should increase, while noise is expected to decrease (unless present in both signals), which in turn results in a higher signal-to-noise ratio. Hence, it might turn out that the signal quality from the abdomen and thorax is quite poor separately, yet still very good combined. We, therefore, in addition to the raw abdominal and thoracic signals, also calculate and measure the signal quality of the $\mathrm{RIP_{sum}}$ signal. Calibration of the belts is rarely done in practice, and so we use the uncalibrated version of the $\mathrm{RIP_{sum}}$ signal. We derive the $\mathrm{RIP_{sum}}$ signal from the *raw* thoracic and abdominal signals using the formula in Equation 5.1 before any preprocessing is applied. Since the unit of measurement is very different between the different sensors anyway (i.e., maintaining the magnitude is unnecessary), the a and b constants are both set to 1. On a side note, as the Shimmer sensor only captures thoracic and not abdominal effort, it is not possible to derive the $\mathrm{RIP_{sum}}$ signal from this sensor.

$$RIP_{sum} = a \cdot RIP_{thorax} + b \cdot RIP_{abdomen} \tag{5.1}$$

Airflow Integration

The piezoelectric sensor type belt (PZT) used by BITalino captures *airflow* rather than *lung volume* (tidal volume), which is captured by both the NOX (RIP) and the Shimmer (IP) sensors. Airflow and tidal volume are both instances of the same physiological process, although from slightly different perspectives. For the signals to be comparable, they need to be transformed into the same unit of measure. As we are interested in breath amplitude for the quality metrics, the unit of tidal volume is what we need. Therefore, we calculate the *cumulative integral* of the signal produced by the BITalino sensors to transform it from *flow* to *volume*. An example can be seen in Figure 5.1, where one may notice how much better the integrated signal resembles the gold standard compared to the raw airflow signal.

One must note that the integration of airflow is not 100% accurate, as it is only an approximation. Furthermore, any noise, and especially baseline wander, have a remarkable effect on the integrated result. Any baseline wander present in the signal accumulates and amplifies as a result of the integration. There are methods that can be used to reduce the presence of baseline wander, such as a high-pass filter or by means of fitting a low-order polynomial (detrending), which we describe further below.

There are multiple integral approximation algorithms available, and two popular ones are the *Trapezoidal Rule* and *Simpson's Rule*. Their precision varies somewhat, with the Simpson's Rule being regarded as more accurate. The Trapezoidal Rule yields an exact answer for polynomials of the first degree, the Simpson's Rule yields an exact answer for polynomials of the third degree or less, and both give an approximation for anything else. In the case of respiratory effort signals, the captured data from the sensors are discrete samples rather than continuous polynomials. This means that the Trapezoidal Rule calculates the result as if linear interpolation is applied to each sample, whereas the Simpson's Rule as if either quadratic or cubic interpolation is applied. As the sampling rate of the signals is rather high, the produced waveforms of the cumulative integration from both methods are almost indistinguishable. Which one of them we choose does, therefore, not matter too much. There is, however, a drawback with the Simpson's Rule as it requires an even number of intervals (i.e., an odd number of samples). This means that if we are to use the Simpson's Rule, the result would be downsampled to half its sampling rate as we can only cumulatively integrate every other sample. Due to this fact and the little difference in accuracy, we have decided to use the Trapezoidal Rule for the cumulative integration of airflow.

Reducing Baseline Wander

The *baseline* of a respiratory effort signal capturing tidal volume is the value on the y-axis in which a breath starts and ends. When this value increases or decreases between breaths, the baseline *wanders*. This is not restricted to different breaths only. Baseline wander can just as well affect a single breath, in which case the breath's start and end y-value differ. Although, this definition does assume that the subject is inhaling and exhaling precisely the same volume every time, which means that in reality, there will be some variations that are not caused by baseline wander. Baseline wander is more generally described as a *trend*, and thus, the correction of baseline wander is commonly referred to as *detrending*.

A high-pass filter can be used to correct or improve baseline wander, as it is essentially just low-frequency noise in the signal. The drawback of a high-pass filter is that it might just as well attenuate parts of the respiratory component itself if the cut-off frequency is not chosen properly. An alternative to a high-pass filter is to fit a low-order polynomial to the signal (i.e., finding the trend), and then subtract it from the signal (detrending). This technique is rather simple and leaves the respiratory component mostly intact. An example can be seen in Figure 5.2, where the ”raw” signal is the result of an integrated airflow signal, which is corrected by subtracting a fitted polynomial of degree 8. This method, however, is less effective when the baseline wander frequency is close to the respiratory component itself (i.e., sharp turns, etc.). In such cases, a high-pass filter might be the better option.

Because integration is so affected by baseline wander, we need a method to correct it. We have decided to use the method of subtracting a fitted low-order polynomial whenever it

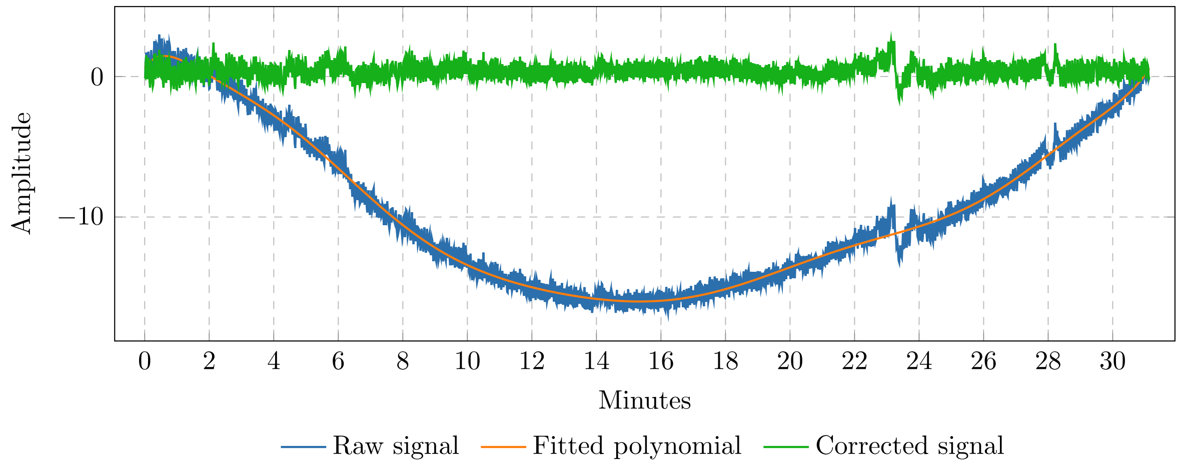


Figure 5.2: Example of heavy baseline wander corrected by fitting a polynomial

is sufficient. The main reason is that it is rather simple yet effective, without affecting the respiratory component too much. In cases where the signal is heavily contaminated by noise and the frequency of the baseline wander is higher, we resort to a high-pass filter instead. As the respiratory component mostly resides in the 0.1–0.5 Hz frequency range, the chosen cut-off frequency should preferably be less than 0.1 Hz.

Synchronization

Before the signals from different sensors can be compared, they need to be synchronized. Neither one of the platforms we possess have any built-in support for hardware synchronization, and as such, the synchronization has to be done in software as a part of the preprocessing. As presented in Section 3.2.1, the use of cross-correlation for synchronization is a viable option, but it does, however, come with a requirement. The waveforms of the two signals must be most similar at the correct point of synchronization. If, for example, one of the signals is heavily contaminated by noise, this may not hold. A solution can be achieved by introducing an event (synchronization point) in the real world which significantly affects the signal (i.e., a huge peak in the data). Such a peak affects the distance between the signals drastically such that the smallest distance is at the point where the peaks from both signals align perfectly. For respiratory effort sensors, on the other hand, this may not even be needed as the breaths themselves are rather distinct features of the signals. This holds as long as the breaths remain the dominant component of the signal and the breath amplitudes between the signals are somewhat correlated. Special care regarding baseline wander must be taken whenever its amplitude becomes greater than the breath amplitudes.

The synchronization is not required to be 100% exact for the metrics we are calculating. The only requirement we have on the synchronization precision is that the peaks of the real breaths captured by the target sensor should be aligned such that they are located somewhere between the start and end of the corresponding breaths in the gold standard. Some synchronization imprecision does, in other words, not matter for the calculation of the signal quality metrics. After performing some initial testing, we find that the use of cross-correlation for synchronizing respiratory effort signals proves to be very accurate, even without the introduction of a synchronization point. We have, therefore, decided to use cross-correlation for the automatic synchronization of the signals. The synchronization result is, however, validated manually for confirmation.

The formula for cross-correlation can be seen in Equation 5.2, where $\hat{y}$ is the first signal, and y is the second signal (i.e., their y-values without the time dimension). The output of this formula is the displacement d^* (in units of samples) between the signals. The synchronization is done by shifting one of the signals along their x-axis in relation to this displacement. When the signals are synchronized, we cut the signals at each end based on their time dimension such that their length becomes equal.

$$d^* = \underset{d \,\in\, \mathbb{Z}}{\arg\max}\left(\sum_{i=-\infty}^{+\infty} \hat{y}[i]y[i+d]\right) \tag{5.2}$$

Sampling Rate

For this definition of cross-correlation to work correctly, the signals must have an equal sampling rate. According to Tripathi (2008), the recommended sampling rate for abdominal and thoracic movement sensors is 100 Hz, and the minimum sampling rate is 25 Hz. For the BITalino sensor, there are four sampling rates to choose from: 1 Hz, 10 Hz, 100 Hz, or 1000 Hz. For the NOX sensor, however, the sampling rate is fixed at 20 Hz for the RIP sensor, and cannot be changed. For the Shimmer sensor, the sampling rate can be freely set to anything up to 2048 Hz. However, the technology used by Shimmer to capture respiration requires a minimum sampling rate of 204.8 Hz. As a result of these limitations, an equal sampling rate amongst the sensors cannot be set in hardware, and thus, have to be set in software instead.

As described in Section 3.2.1, a decimation (or downsampling) is regarded as being more accurate compared to an upsampling. It is, however, not desirable to set the sampling rates arbitrarily high as this increases the data load considerably, and is especially unnecessary if the signals are to be decimated in software right away anyway. Based on this information, we have decided to record the BITalino signal with a sampling rate of 100 Hz, and the Shimmer sensor at 512 Hz. We decimate these signals to 20 Hz in the preprocessing phase for them to become equal to the NOX's fixed sampling rate.

Standardization

Standardization is an alternative to normalization. Instead of scaling the data to a specific range, its mean and standard deviation are set to *zero* and *one*, respectively. The formula for standardization is shown in Equation 5.3, where X are the samples of the signal, μ is the signal's mean value, and σ is the signal's standard deviation. The main advantage of standardization over normalization is that normalization is heavily affected by outliers, whereas standardization is rather robust. For example, given two different signals, both with a steady breath amplitude of *two* (relative value). If one of the signals contains an outlier value of 100, the breaths from this signal would after normalization become so small compared to the breaths in the other signal that a visual waveform comparison would be impossible. With standardization, on the other hand, this outlier would have no noticeable effect on the scale of the breaths (assuming the signals are of sufficient length).

$$X_{new} = \frac{X - \mu}{\sigma} \tag{5.3}$$

None of the preprocessing steps, nor any of the quality metrics, are dependent on normalization or standardization. However, because the raw values produced by each of the sensors are so different, they need to be scaled to be visually comparable. We have, therefore, decided to standardize the signals as part of the preprocessing phase. Outliers in these kinds of signals are so common that normalization for the sake of visual comparability is mostly meaningless.

The Preprocessing Steps

To summarize the preprocessing steps, we perform the following actions prior to any metric calculations:

1. Calculate the uncalibrated RIP_{sum} signal.
2. Remove baseline wander.
3. Integrate airflow signal to volume.
4. Resample the signals to 20 Hz.
5. Synchronize with the use of *cross-correlation*.
6. Standardize the signals.

5.1.2 Metrics

Definition of a Breath

As previously stated, a breath as recorded by respiratory effort sensors is by definition the same as a peak, constrained by physical limitations. In simple terms, the *peaks* of a signal can be defined as $peaks = \{s \in S \mid s_{i-1} < s_i > s_{i+1}\}$, where S is the signal. In other words, a peak is any sample that is higher than its immediate neighbors. However, this definition yields many peaks that are not breaths (especially in noisy signals) and requires careful filtering according to our requirements. Additionally, it must also be corrected to support *flat* peaks (i.e., a peak may span multiple samples). Instead of "reinventing the wheel," we extract the peaks by using the *findpeaks* function from the *MATLAB* library (The Mathworks, Inc. 2016a), which does most of this filtering automatically. The exact parameters supplied for this function are described further in Section 6.5.2.

To derive the *start* and *end* point of a breath, we follow the same definition as proposed by Retory et al. (2016). The start of a breath is defined as the *minimum* value between the breath and the preceding breath, while the end as the minimum value between the breath and the succeeding breath. In other words, the end of one breath is the start of another. The formula for start and end can be seen in Equation 5.4 and Equation 5.5, where S is the signal, i is the index in S of the preceding breath, j is the index of

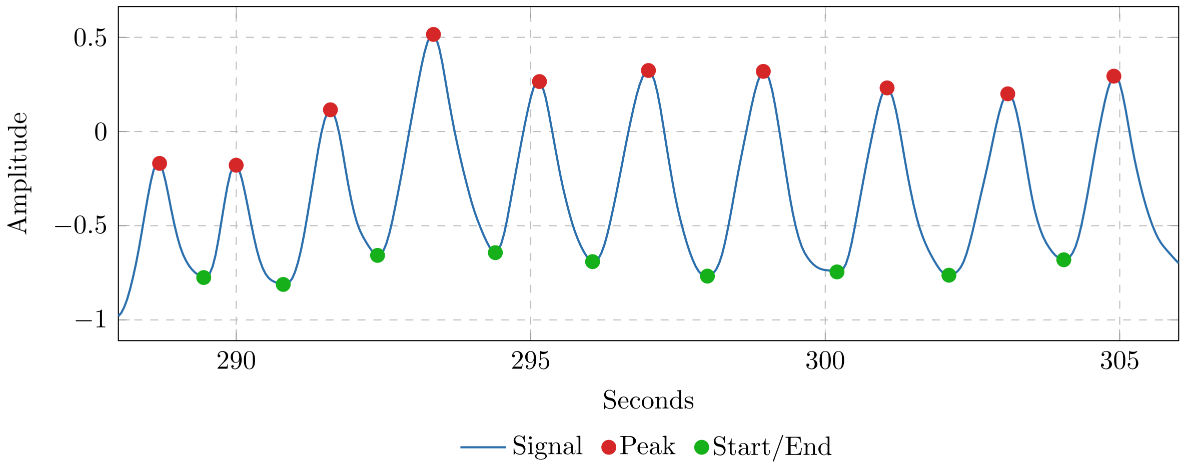


Figure 5.3: Start and end of a breath are defined as the minimum between two peaks

the target breath, and k is the index of the succeeding breath (i.e., their peaks). See Figure 5.3 for a visual illustration. The duration of a breath is measured as the time distance between its start and end. We specified a requirement regarding the duration of breaths in Section 4.2, namely that breaths should be no longer than twelve seconds and no shorter than 0.6 seconds. There is, however, an issue with this definition of breath duration. In the presence of breathing stops (gaps), the breaths at the start and end of the gap may get a duration which is longer than twelve seconds, even though their real duration is not. Due to this issue, we do not filter out breaths that are longer than twelve seconds, but instead, limit the distance the start/end of a breath can be from its peak. In other words, the interval in which we find the minimum value (start/end), is either the distance from the peak to either neighboring peaks or six seconds from the peak in either direction (peak is in the middle of the breath, $\frac{12}{2} = 6$). If this causes the peak's amplitude to get too low, the breath is filtered out. On a side note, the breaths at the boundaries of the signal are very likely to not get their *true* start and end value by this method, and hence not their true amplitude either. These breaths may, therefore, be excluded from the quality measurement altogether to get a more fair quality comparison.

$$start = min(S_i, S_{i+1}, S_{i+2}, ..., S_j) \tag{5.4}$$

$$end = min(S_j, S_{j+1}, S_{j+2}, ..., S_k) \tag{5.5}$$

Since these kinds of signals are very commonly affected by a varying degree of baseline wander, measuring the amplitude of a breath based on its peak y-value is unreliable. The amplitude of a breath should, therefore, be measured as the difference between its *peak* and its *start/end* value. As the start and end value of a breath may vary, we calculate the breath amplitude as the difference from its peak to the *mean* of the start/end values (see Equation 5.6). After conducting some initial testing, we find that this definition of the amplitude is more accurate (it is closer to the gold standard) compared to taking either the minimum (Equation 5.7) or maximum distance. However, it has a significant drawback in the context of disregarding peaks with an amplitude lower than 10% of the mean breath amplitude. When the distance from the peak to either its start or end is very close to zero, its amplitude may still be way above the threshold with this definition. As a result, we use an alternative amplitude definition for the peak filtering. This definition can be seen in Equation 5.7, where the amplitude is the minimum distance from its peak to either its start or end.

$$amplitude = peak - \frac{start + end}{2} \tag{5.6}$$

$$amplitude = min(peak - start, peak - end) \tag{5.7}$$

One of the requirements we set for the breath detection is that a peak has to be at least 10% as high as the mean breath amplitude to be regarded as a breath (Section 4.2). The challenge with this is to determine the mean breath amplitude *before* detecting the breaths themselves. This is rather hard (if not impossible) to do accurately, and so we must use an approximation instead. We derive this approximation by splitting the signal into non-overlapping tumbling windows, from which we take the maximum amplitude ($max - min$ y-value). The *mean* of these amplitudes is the approximated *mean breath amplitude*. See Equation 5.8, where S is the signal, and w is the window width in the number of samples. We set the window width to be *four* seconds wide for this approximation. The reason is that four seconds is approximately the duration of a breath during normal breathing (which is *12–18* breaths per minute for adults). To give an example of the precision of this approximation method, we got an approximated value of *2.6636*, when the real value was *2.6648*. The precision of this approximation depends on either how close the respiration rate in the signal is to four seconds or the stability of the breath amplitudes. If the breath amplitudes are somewhat stable, a wider window width is tolerated without losing precision. With this approximation as a starting point, an iterative approach can be applied to increase the accuracy of the estimate further if desired. The accuracy of the automatic breath detection is, however, not the primary focus of this thesis. We manually validate and confirm the process anyway, and automatic detection is used only as a means to speed up the process. This breath amplitude approximation method is, therefore, very much adequate for our purpose.

$$\begin{aligned} n &= \frac{|S|}{w} \\ amplitude_{mean} &= \frac{1}{n}\sum_{i=0}^{n} max(S_{iw}, S_{iw+1}, ..., S_{iw+w-1}) - min(S_{iw}, S_{iw+1}, ..., S_{iw+w-1}) \end{aligned} \tag{5.8}$$

Breath Detection Accuracy

We decided in Section 4.2 that the signal quality of a sensor regarding its ability to detect apneas should be measured by its ability to correctly identify breaths. Furthermore, the interpretation of the quality metric should be trivial in relation to how good the sensor is at detecting apneic events. Based on these requirements, we have decided to use the *sensitivity* and *positive predictive value* metrics as our primary breath detection accuracy metrics. See Equation 5.9 and Equation 5.10, where $|B_{true}|$ is the number of correctly identified breaths, $|B|$ is the number of all detected breaths, and $|R|$ is the number of real breaths (as detected by the gold standard). For a breath to be regarded as true, its peak has to be between the start and end of a real breath. See Equation 5.11, where B are the breaths detected by the target sensor, and R the breaths detected by the gold standard. The sensitivity of a sensor directly reflects the sensor's ability to identify real breaths, while the positive predictive value yields information about the proportion of false breaths. What these metrics lack, however, is information about the distribution of the errors/artifacts. As such, we additionally use the *clean minute proportion* metric. See Equation 5.12, where $|M_{clean}|$ is the number of *clean* minutes, and $|M|$ is the total number of minutes in the signal. A minute is regarded as clean if both the sensitivity and positive predictive value are 100% during the minute. See Equation 5.13, where M are all minutes of the signal, s yields the sensitivity, and *ppv* yields the positive predictive value. If the errors/artifacts of a sensor are only present once in a while, the proportion of accurate minutes could still be high, and thus the sensor might still be very much usable.

$$Sensitivity = \frac{|B_{true}|}{|R|} \times 100\% \tag{5.9}$$

$$Positive\ predictive\ value = \frac{|B_{true}|}{|B|} \times 100\% \tag{5.10}$$

$$B_{true} = \{b \in B \mid r_{start} < b_{peak} < r_{end},\ r \in R\} \tag{5.11}$$

$$Clean\ minute\ proportion = \frac{|M_{clean}|}{|M|} \times 100\% \tag{5.12}$$

$$M_{clean} = \{m \in M \mid s(m) = 100\ and\ ppv(m) = 100\} \tag{5.13}$$

Breath Amplitude Accuracy

How good a sensor is at detecting hypopneas depends on how accurate the amplitudes of the detected breaths are. As such, we measure the signal quality with respect to hypopnea detection by comparing the breath amplitudes of the target sensor with the breath amplitudes of the gold standard. Because the different sensors are using different scales, their amplitude values cannot be compared directly. Even common normalization/standardization techniques are not sufficient because of the presence of noise, baseline wander, and outliers. Instead, the *relationship* between the amplitudes from both sensors needs to be determined. As explained in Section 4.1.2, this relationship should be *linear*, and as such, is obtained through linear regression. The distance from a breath's amplitude value to the corresponding value on the regression line is its relative error value.

After the error values of all the breath amplitudes are determined, they must be summarized to represent the overall signal quality of the sensor. There are many statistical methods that can be used to summarize these error values, such as *RMSE*, *coefficient of determination*, *Spearman's correlation coefficient*, etc. (see Section 3.4). However, as the presence of hypopneic events is identified based on the *relative percentage reduction* of the breath amplitudes in relation to the baseline breath amplitude, none of these methods yields a trivial interpretation. For example, given an *RMSE* error value of *0.3*. One cannot know if this is a good or bad quality sensor in relation to hypopnea detection. To be able to reason about that, one would have to additionally know the baseline breath amplitude. Consequently, the accuracy metric between different sensors, or even between different signal captures from the same sensor, cannot be compared because the baseline breath amplitude may vary. One must also remember that the mere *existence* of a relationship is not of interest. The relationship is assumed to exist, and it is the *variance* that is of importance. A better option is, therefore, to calculate the *mean percentage error*. An error rate of, for example, 10% can immediately be interpreted as *the breath amplitudes are on average 10% off in either direction.*

We have, therefore, decided to use the *Weighted Absolute Percentage Error* (*WAPE*) (see Section 3.4) metric to represent the breath amplitude accuracy. The formula for *WAPE* can be seen in Equation 5.14, where E are the values at the regression line, and B are the actual breath amplitudes as recorded by the target sensor. The expanded formula shown in Equation 5.15 may be easier to understand conceptually, where $\overline{B}$ is the mean of the breath amplitudes. The way this metric works is that we regard the *mean breath amplitude* as being the baseline, and then calculate the reduction/increase as a percentage difference from this baseline. For example, let the baseline be *1*, and the value at the regression line be *0.4* (i.e., a 60% reduction from the baseline). If a sensor detects the amplitude of this breath as *0.5*, the error would be 10% ($\frac{|0.4-0.5|}{1} \times 100\%$).

$$\begin{gathered} n = |B| \\ WAPE = \frac{\sum\limits_{i=0}^{n} |E_i - B_i|}{\sum\limits_{i=0}^{n} B_i} \times 100\% \end{gathered} \tag{5.14}$$

$$\begin{gathered} n = |B| \\ WAPE = \frac{1}{n} \sum_{i=0}^{n} \frac{|E_i - B_i|}{\overline{B}} \times 100\% \end{gathered} \tag{5.15}$$

For this metric to work, we need to determine the linear relationship between the breath amplitudes and derive the regression line. There are many linear regression algorithms available, with one of the most common being the *(ordinary) least squares* (OLS) algorithm. The OLS algorithm fits a regression line such that the squared error is as low as possible. Consequently, if there are strong outliers in the data, an otherwise obvious relationship is not captured by this algorithm. See Figure 5.4 for a visual illustration. In this example, the target sensor is unable to correctly detect the higher breath amplitudes, but the linear relationship is still present for the lower-middle amplitude breaths. Such a deviation from the relationship should be penalized, and thus, an outlier robust regression algorithm is more appropriate in this case compared to the OLS algorithm.

Two popular outlier robust regression algorithms are *Theil-Sen* and *Random Sample Consensus* (RANSAC). Put shortly, the Theil-Sen algorithm finds the slope between all pairs of points and then chooses the median of these slopes as the model. This can be computed exactly with a complexity of $O(n^2)$. The amount of data we are working

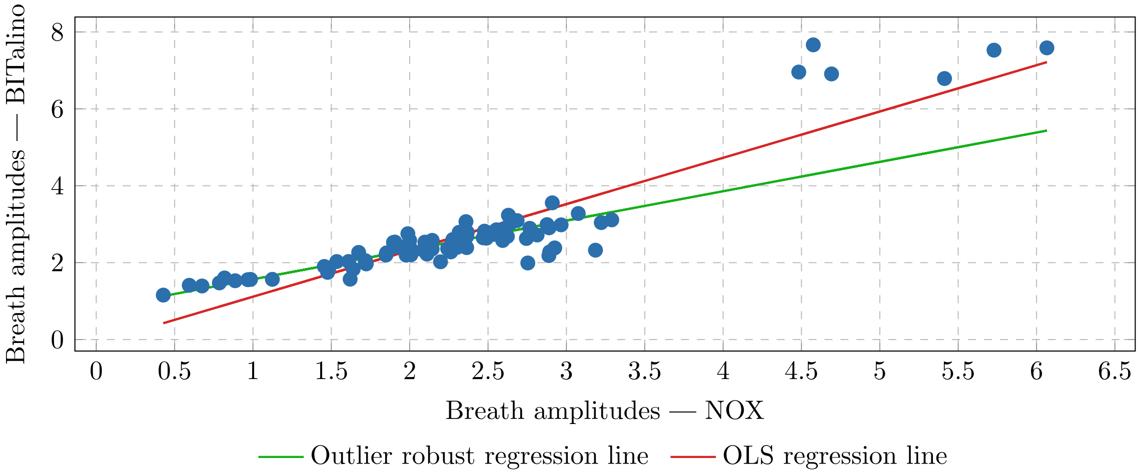


Figure 5.4: OLS vs outlier robust regression algorithm

with is not that vast considering that a breath often spans multiple seconds, and so the complexity of this algorithm is not of significant concern. The RANSAC algorithm works by randomly picking the minimum number of samples required to fit a model, and then uses a voting scheme where any data point that fits this model is regarded as an *inlier*. These two steps are repeated iteratively until the number of inliers of a model exceeds a threshold. Both of these algorithms yield very similar results for our case, and both are much better than the OLS algorithm. The RANSAC algorithm, however, produces a slightly different regression line every time it is run on the same data because of its random nature. This causes the accuracy metric to vary ever so slightly between different runs, which is not desirable. Because of this factor, we have decided to use the Theil-Sen regression algorithm for the metric calculation.

There are at least three aspects of the WAPE metric that should be emphasized. Firstly, if the slope of the regression line is exactly zero, then the metric is misleading/wrong. This may happen if the breath amplitudes are constant or vary evenly around a point. When the slope is zero, it is impossible to transfer (calibrate) the target sensor to the same unit of measure as the gold standard (or absolute units). Secondly, the result derived from random data with an even distribution is 50% regardless of the slope of the regression line. Thirdly, the breath amplitudes (once extracted from the signal) must *not* be normalized/standardized. The reason is that the (intra) ratio between the amplitudes of the breaths must not be altered. For example, let the mean breath amplitude be 1.5, and the amplitude of a sample breath be 1.2 (i.e., a 20% reduction from the mean). If these breath amplitudes are normalized such that the mean breath amplitude becomes 0.5 and the sample breath amplitude becomes 0.2, then their relative ratio changes ($\frac{1.2}{1.5} \neq \frac{0.2}{0.5}$, suddenly a 40% reduction).

5.2 Preliminary Testing

Signal capture sessions performed overnight while the subjects are asleep are with no doubt the most representative sessions for sleep apnea monitoring. These longer sessions, however, require significantly more work, both with respect to execution as well as subject recruitment, compared to shorter sessions that can be performed in a laboratory during wakefulness. Shorter sessions are, thus, preferred over longer overnight sessions as long as they prove to be sufficiently representative (see Section 4.3.2). To determine if shorter sessions performed during wakefulness indeed are sufficient to measure the signal quality of the sensors, we perform several preliminary tests. The main purpose of this preliminary testing is more specifically to determine if and how the signal quality from the sensors changes over time and how changes in body position affect the signal. In addition, these preliminary tests may also wind up uncovering sensor specific oddities and traits that should be taken into account when we design the signal capture procedure.

5.2.1 Tests

These preliminary tests focus primarily on the BITalino sensor. The reason is mainly that the Shimmer sensor is certified as medical grade equipment, and its quality is, therefore, expected to be rather good overall. That said, the Shimmer sensor is not excluded from these preliminary tests altogether, but it is not tested as extensively as the BITalino sensor. Also, the RespiBAN and FLOW sensors are not included because we acquired them too late in the process. For the BITalino sensor, we perform the preliminary tests over several sessions, some in which the subject is awake, while others in which the subject is sleeping. To measure how the signal quality changes over time, we split each signal capture into smaller chunks and then compare these on an intra-capture basis. We measure the quality of these captures using the methods described in Section 5.1, namely by means of *breath detection accuracy* and *breath amplitude accuracy*. The RIP_{sum} signal is excluded from the preliminary testing because the mere signal quality in these tests are not of interest, but rather how the sensors *behave*.

We perform the following preliminary tests:

- Four 30-minute tests, capturing both abdominal and thoracic breathing, sitting relatively still in a chair.
- Two 2.5-hour tests, capturing both abdominal and thoracic breathing, at night while the subject is sleeping.
- Various smaller experiments to uncover:
  - How sensor entrapment affects the signal (e.g., by lying on the sensor).
  - How minor belt misplacement affects the signal.
  - How the technology used by Shimmer differs from the belt type sensors.

5.2.2 Findings

Sensor Initialization

One of the first things we discovered is that the first two minutes of each signal is somewhat noisy. This applies to both the BITalino sensor as well as the gold standard (NOX). We suspect that this may be due to a combination of sensor initialization/calibration and movement by the subject. When the respiratory effort belt is strapped around a new subject, it has to adapt to a new baseline circumference. An example of this kind of noise can be seen in Figure 5.5. Notice how the signal is very noisy at the start and stabilizes

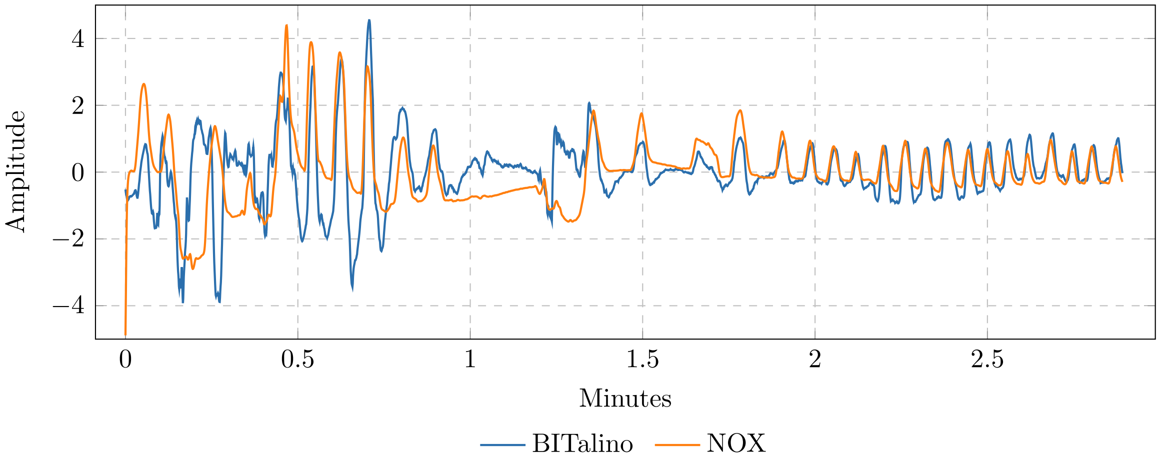


Figure 5.5: First three minutes of a signal capture

towards the end. Because of this phenomenon, it may be a good idea to exclude the first two minutes from the quality measurement altogether.

30-minute Captures

For the 30-minute captures, we removed the first two minutes of each capture and then split the signal into four chunks of about seven minutes each. In other words, *Part 1* consists of minute 2–8, *Part 2* of minute 9–15, and so on. This makes it possible to see how the signal quality differs between each part, to get an idea of how/if the signal quality changes as time passes. The results for the breath detection accuracy can be seen in Table 5.1, and the results for the breath amplitude accuracy can be seen in Table 5.2.

After a quick visual inspection of the signal captures, the first impression is that the overall signal quality remains somewhat consistent over time. By studying the results, it becomes clear that the breath detection accuracy does indeed not keep changing in any one direction over time. It is rather stable with minor fluctuations between each part. For three out of eight captures, *Part 4* does actually have a better PPV compared to *Part 1*. Moreover, there is possibly a trend where the PPV decreases until somewhere between *Part 2 and 3* before it starts increasing again. This may indicate that a capture of about fifteen minutes (*Part 1–2*) is sufficiently long enough to capture the average breath detection accuracy for the given sensor. One may notice that the sensitivity is often somewhat good, whereas the PPV is generally a little lower. This suggests that it is mainly *false breaths* that are of concern; at least for these captures.

For the breath amplitude accuracy, on the other hand, there is possibly a slight decrease in quality over time for most of the captures. *Part 4* has, in fact, a lower breath amplitude accuracy compared to *Part 1*, for seven out of eight captures; although with varying degree of recession. The difference in accuracy between each part might be the result of changes in body position, causing motion artifacts or amplitude changes (caused by changes in belt tightness). Nonetheless, this becomes more clear from the longer overnight sessions.

In general, the abdominal signal (slightly more often than not) have a higher accuracy overall compared to the thoracic signal. This is most likely because the breathing style in

	Part 1 (min. 2–8)	Part 2 (min. 9–15)	Part 3 (min. 16–22)	Part 4 (min. 23–29)
Capture 1 (abdomen)	S: 100.00% PPV: 99.14% CMP: 87.50%	S: 100.00% PPV: 98.86% CMP: 87.50%	S: 98.84% PPV: 100.00% CMP: 87.50%	S: 100.00% PPV: 100.00% CMP: 100.00%
Capture 1 (thorax)	S: 100.00% PPV: 82.73% CMP: 12.50%	S: 100.00% PPV: 98.88% CMP: 87.50%	S: 97.73% PPV: 91.49% CMP: 50.00%	S: 100.00% PPV: 94.06% CMP: 50.00%
Capture 2 (abdomen)	S: 100.00% PPV: 100.00% CMP: 100.00%	S: 100.00% PPV: 98.17% CMP: 71.43%	S: 99.07% PPV: 99.07% CMP: 71.43%	S: 100.00% PPV: 97.73% CMP: 85.71%
Capture 2 (thorax)	S: 100.00% PPV: 100.00% CMP: 100.00%	S: 98.18% PPV: 100.00% CMP: 71.43%	S: 96.33 PPV: 97.22% CMP: 28.57%	S: 95.35% PPV: 95.35% CMP: 57.14%
Capture 3 (abdomen)	S: 100.00% PPV: 93.22% CMP: 57.14%	S: 100.00% PPV: 83.85% CMP: 0.00%	S: 99.07% PPV: 84.92% CMP: 28.57%	S: 99.10% PPV: 95.65% CMP: 57.14%
Capture 3 (thorax)	S: 100.00% PPV: 84.62% CMP: 14.29%	S: 99.08% PPV: 68.35% CMP: 0.00%	S: 100.00% PPV: 63.69% CMP: 14.29%	S: 99.09% PPV: 66.46% CMP: 0.00%
Capture 4 (abdomen)	S: 98.15% PPV: 100.00% CMP: 87.50%	S: 98.28% PPV: 100.00% CMP: 75.00%	S: 88.60% PPV: 93.52% CMP: 0.00%	S: 80.77% PPV: 100.00% CMP: 50.00%
Capture 4 (thorax)	S: 97.22% PPV: 95.45% CMP: 37.50%	S: 100.00% PPV: 98.31% CMP: 75.00%	S: 95.58% PPV: 95.58% CMP: 37.50%	S: 98.13% PPV: 99.06% CMP: 75.00%

** S: sensitivity, PPV: positive predictive value, CMP: clean minute proportion*

Table 5.1: Breath detection accuracy for the 30-minute captures

	Part 1 (min. 2–8)	Part 2 (min. 9–15)	Part 3 (min. 16–22)	Part 4 (min. 23–29)
Capture 1 (abdomen)	9.70%	13.94%	15.68%	16.29%
Capture 1 (thorax)	23.79%	27.71%	18.61%	33.75%
Capture 2 (abdomen)	10.66%	14.03%	18.76%	22.60%
Capture 2 (thorax)	15.16%	13.45%	22.45%	26.76%
Capture 3 (abdomen)	25.68%	25.19%	27.79%	27.48%
Capture 3 (thorax)	18.68%	24.71%	23.73%	29.92%
Capture 4 (abdomen)	15.82%	26.97%	64.78%	23.89%
Capture 4 (thorax)	32.72%	19.18%	33.39%	28.17%

** mean amplitude percentage error (WAPE)*

Table 5.2: Breath amplitude accuracy for the 30-minute captures

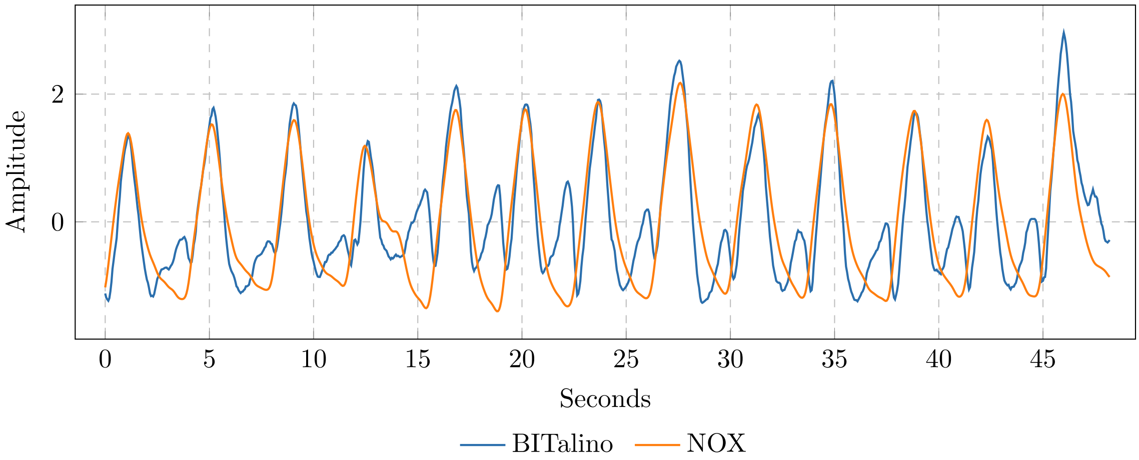


Figure 5.6: False breaths between real breaths

these captures are more prominent in the abdomen, causing a higher breath amplitude, and thus a higher *signal-to-noise* ratio.

One may notice that although the PPV is much lower in *Capture 3 (thorax)* compared to the other captures, its breath amplitude accuracy is still about average. This happens because the amplitude accuracy is only calculated based on correctly identified breaths (i.e., breaths matched with the gold standard). In other words, *Capture 3 (thorax)* contains a lot of false breaths which are not taken into account when calculating the breath amplitude accuracy. An example can be seen in Figure 5.6, where the BITalino signal exhibits high peaks (which are often misinterpreted as breaths) between each real breath.

In *Capture 4 (abdomen)*, we discovered an oddity where the breath amplitudes suddenly dropped over a period, without any changes to breathing intensity or body position (see Figure 5.7). This effect is clearly visible in both the sensitivity and breath amplitude accuracy metrics. These periods are scattered throughout the capture, and last from about 30 seconds up to about three minutes each. This phenomenon has yet only been observed for that specific signal capture (the thoracic signal does not exhibit this phenomenon), however, multiple times during the capture. The cause is yet unknown, but it seems to be inherent to the belt itself as no external adjustments were made to either the breathing, body, or belt during the capture. How common this kind of behavior is will show as we gather more captures, but it is, nevertheless, clear that such behavior has a direct negative impact on both the breath detection as well as breath amplitude accuracy. It is expected that even trained personnel would annotate these periods as at least hypopneic, if not even as apneic.

Overnight Captures

After a visual inspection of the overnight captures, it is evident that the breath amplitudes vary significantly between different sleeping positions for the BITalino sensor compared to the NOX sensor (see Figure 5.8 and Figure 5.9). For the BITalino capture (Figure 5.8), the periods with high amplitudes are from a duration of sleeping on the side, whereas the lower amplitude periods are from a duration of sleeping in the supine (back) position.

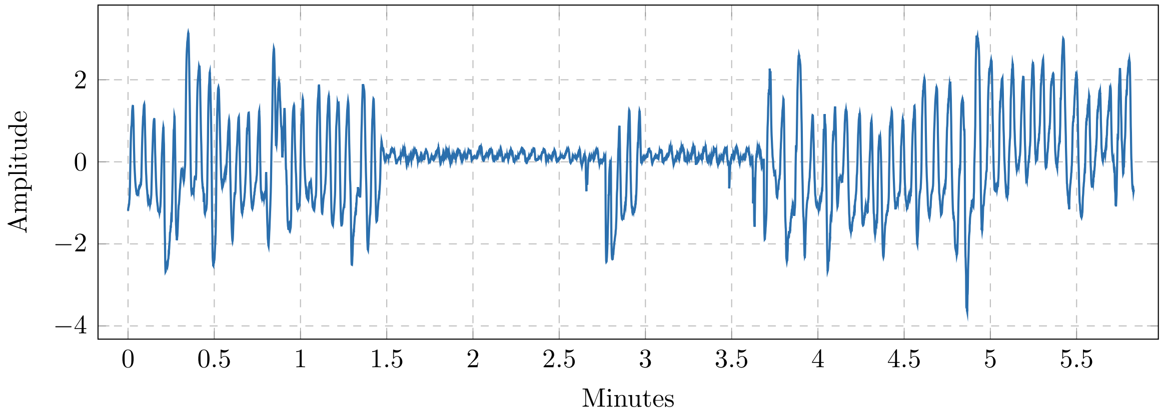


Figure 5.7: Capture 4 (abdomen) — sudden periods of low amplitude

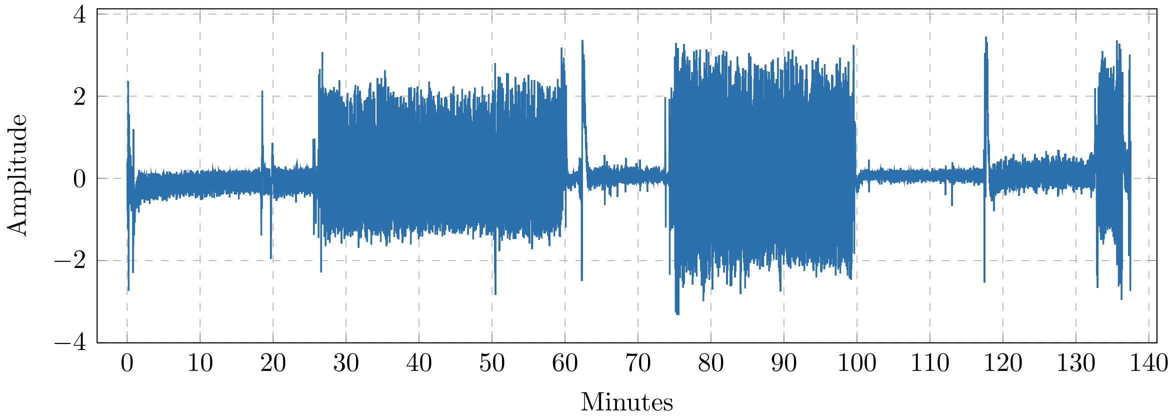


Figure 5.8: Overview of an overnight capture from BITalino

As seen from the NOX capture (Figure 5.9), the breath amplitudes are somewhat stable across sleeping positions, however, with some spikes as a result of motion artifacts (e.g., by the change of sleeping position). These examples are taken from an abdominal belt signal, but the same phenomenon can also be clearly seen from the thoracic signals, although to a slightly lesser degree.

With such a massive variation in baseline breath amplitude between different sleeping positions, a challenge arises. The periods with different baseline breath amplitude either need to be compared with the gold standard separately, or careful adaptive normalization techniques must be applied. The breath amplitude relationship would otherwise be completely misleading (see Figure 5.10). More interestingly, one can in Figure 5.10 make out at least three distinct clusters, which indicates that at least three such periods (with different baseline breath amplitudes) exist in the data. In this particular instance, one can even quite easily make out visually which cluster corresponds to which period. Another noteworthy observation is that even periods with the same sleeping position have a big enough difference in baseline breath amplitude to form distinct clusters.

Another oddity we discovered is that the BITalino signal may suddenly invert its values across the y-axis for a period of time. In other words, the signal is flipped, which means that as the belt distraction expands, the y-value decreases instead of increases. This

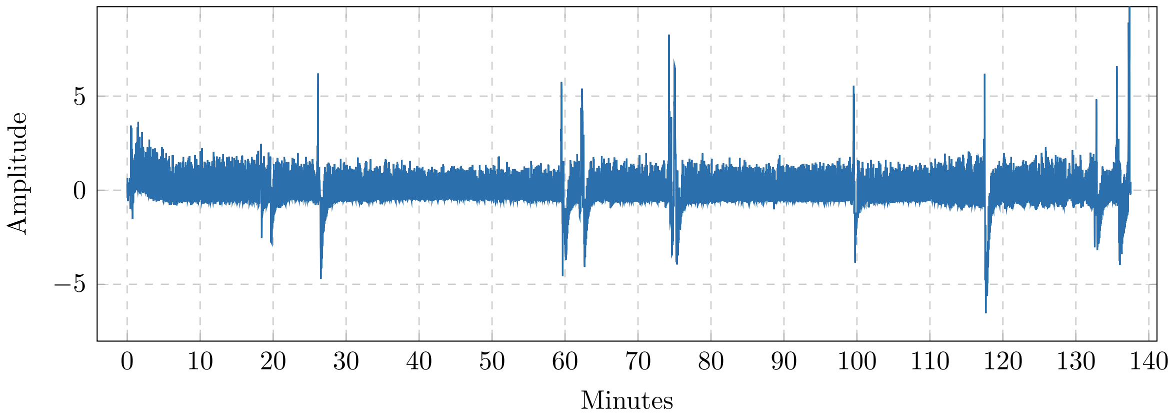


Figure 5.9: Overview of an overnight capture from NOX

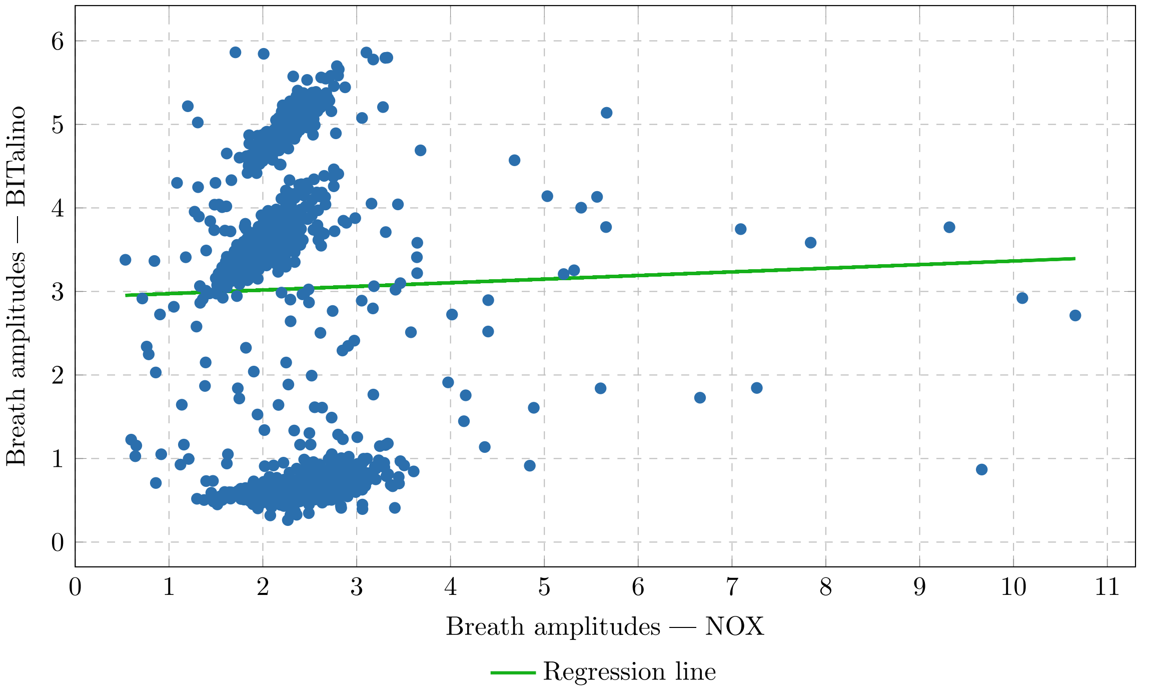


Figure 5.10: Breath amplitude relationship when whole signal is standardized together

Capture 1: Breath Detection Accuracy

	Supine (min. 3–18)	Side (min. 27–55)	Supine (min. 62–71)	Side (min. 76–96)	Supine (min. 99–113)
Abdomen	S: 100.00% PPV: 100.00% CMP: 100.00%	S: 99.75% PPV: 100.00% CMP: 96.55%	S: 100.00% PPV: 98.18% CMP: 77.78%	S: 98.97% PPV: 100.00% CMP: 95.24%	S: 97.65% PPV: 99.52% CMP: 66.67%
Thorax	S: 98.94% PPV: 75.30% CMP: 13.33%	S: 100.00% PPV: 99.75% CMP: 96.55%	S: 98.15% PPV: 100.00% CMP: 88.89%	S: 98.96% PPV: 99.30% CMP: 90.48%	S: 100.00% PPV: 68.28% CMP: 6.67%

** S: sensitivity, PPV: positive predictive value, CMP: clean minute proportion*

Capture 1: Breath Amplitude Accuracy

	Supine (min. 3–18)	Side (min. 27–55)	Supine (min. 62–71)	Side (min. 76–96)	Supine (min. 99–113)
Abdomen	10.48%	5.89%	19.00%	5.48%	19.77%
Thorax	22.65%	10.96%	14.05%	22.03%	27.44%

** mean amplitude percentage error (WAPE)*

Table 5.3: Breath detection and amplitude accuracy for overnight capture 1

seems to be triggered randomly as the subject changes position, in other words, by physical movement. While this phenomenon is rather easy to correct once identified, it may not be trivial to detect in the first place. The only indication present without comparing it to the gold standard is that the amount of noise is usually higher between breaths compared to during a breath. Moreover, it is unclear how common this kind of behavior is, as it happens multiple times during both overnight captures (even without any suspected movement), but not in any of the 30-minute captures. In the second overnight capture, it happened within five minutes, which is before the subject fell asleep. The motion associated with lying down in bed is what seems to be the trigger in this particular case. This suggests that it should be possible to uncover this phenomenon as easily during shorter wakeful sessions. It may not have been present in the 30-minute captures because no prominent position changes were involved.

To evaluate the quality of the overnight captures, we compare the periods with different baseline breath amplitudes separately. Periods of inverted y-axis are manually identified and corrected on a best effort basis. Very short periods, as well as data captured during position changes, are excluded from the comparison. The results are presented in Table 5.3 and Table 5.4, in the same units as for the 30-minute captures. However, as the time slices and body positions vary between the two overnight captures, they are separated into different tables.

Based on these results, the abdominal signal is again superior compared to the thoracic signal. In addition, the signal quality from sleeping on the side is exceptionally good, especially for the abdominal signal. Regarding any quality changes over time, there is not any noticeable trend. Rather, the quality is mostly affected by position changes causing changes to the baseline breath amplitude. It becomes harder to distinguish noise from breaths as the breath amplitudes become lower.

Capture 2: Breath Detection Accuracy

	Supine (min. 3–20)	Side (min. 22–70)	Side (min. 71–96)	Supine (min. 97–121)	Side (min. 124–168)
Abdomen	S: 99.64% PPV: 99.28% CMP: 85.00%	S: 99.71% PPV: 100.00% CMP: 95.65%	S: 99.14% PPV: 99.71% CMP: 92.00%	S: 90.00% PPV: 98.62% CMP: 33.33%	S: 99.86% PPV: 99.86% CMP: 95.65%
Thorax	S: 96.80% PPV: 91.58% CMP: 30.00%	S: 99.71% PPV: 100.00% CMP: 95.65%	S: 98.85% PPV: 99.71% CMP: 88.00%	S: 96.47% PPV: 100.00% CMP: 79.17%	S: 99.45% PPV: 99.45% CMP: 91.30%

** S: sensitivity, PPV: positive predictive value, CMP: clean minute proportion*

Capture 2: Breath Amplitude Accuracy

	Supine (min. 3–20)	Side (min. 22–70)	Side (min. 71–96)	Supine (min. 97–121)	Side (min. 124–168)
Abdomen	22.65%	6.15%	8.21%	30.78%	12.52%
Thorax	35.86%	9.10%	13.72%	16.67%	8.50%

** mean amplitude percentage error (WAPE)*

Table 5.4: Breath detection and amplitude accuracy for overnight capture 2

Sensor Entrapment Experiments

The sensor part of a piezoelectric respiratory effort belt does not span the whole circumference of the belt, but only a small area. This makes these kinds of sensors prone to entrapment, as the non-sensor part of the belt can expand while the sensor part is kept still (e.g., by lying on the sensor part). Therefore, we perform several smaller experiments to determine the effect this has on the signal quality. These tests involve behavior that can happen at night, for example, lying on the sensor as well as resting the arms directly on top of the sensor.

The results of these tests show that the general outcome of sensor entrapment is a significant decrease in breath amplitudes. This is in contrast to the RIP type belt, which is not affected by entrapment at all. The degree of the amplitude decrease depends on how trapped the sensor becomes. If the amplitudes decrease too much, the breaths become indistinguishable from noise, which in turn renders the signal useless. On a side note, the abdominal signal is slightly less affected compared to the thoracic signal when the subject is lying on the sensors; at least in these tests.

A somewhat strange oddity we discovered when performing these tests was that the BITalino belts malfunctioned from time to time. They would suddenly start to only output values near maximum or minimum (see Figure 5.11), and the only way to restore them was to physically touch/adjust the belts. The movement from the breathing alone is enough to correct the signal. We have experienced this behavior on four different belts, and hence it is not very rare, nor caused by a faulty belt. Although, physical movement a bit stronger than normal breathing (e.g., by lying on the sensor) seems to be the main trigger. If this happens during the night, the whole signal might end up becoming useless. Thus, it is important that we try to determine how commonly this occurs during normal body position changes in bed.

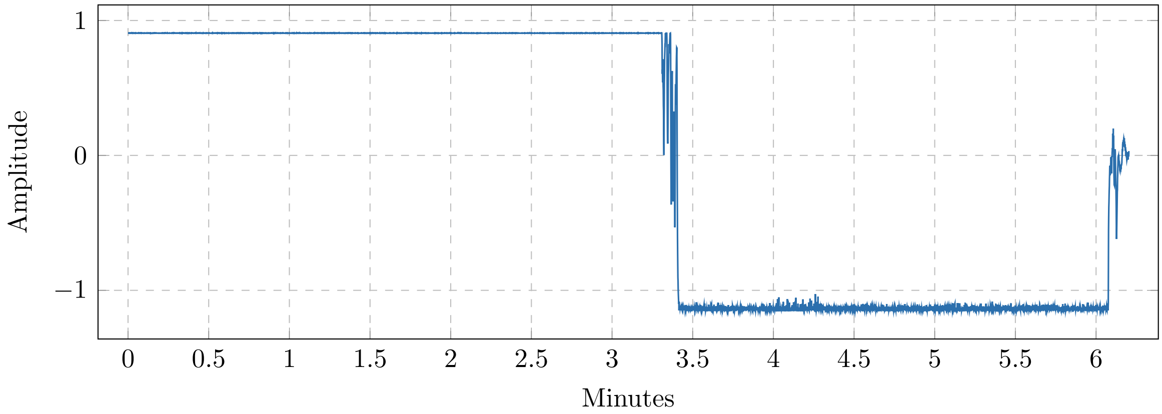


Figure 5.11: BITalino belt malfunctioning

Effect of Belt Placement

The respiratory process can be described as a system with two degrees of freedom of motion (Konno and Mead 1967). This means that to accurately measure tidal volume from body movement, the sum of both the thoracic and abdominal movements are required. Moreover, the belts should be placed at the abdominal and thoracic locations where the movements are most prominent. The recommended positioning of respiratory effort belts is at the level of the umbilicus for the abdominal belt and at nipple level (just above or below for females) for the thoracic belt. Any variations from this potentially reduce the magnitude of the captured breaths, and hence the signal-to-noise ratio declines. In addition to the mere placement of the belts, their tightness also significantly affects the signal quality. The belts should optimally be fit *snugly* around the subject. In other words, the belts should be tight enough to follow every motion associated with breathing while still minimizing motion artifacts, but not too tight either. Whereas a very tight fitted belt might produce higher breath amplitudes, the breath amplitude linearity has been shown to degrade when the belt distraction becomes too high; at least for some piezoelectric belts (Vaughn and Clemmons 2012).

Because we use two sets of belts in our tests, two abdominal and two thoracic belts, a *perfect* fit is not possible. We instead position the belts as close to each other as possible (without interfering), and with the optimal position right in the middle. For example, the level of the umbilicus is right between the two abdominal belts, with a gap between the belts as small as possible. This yields two ordering possibilities, the gold standard can either be placed below or above the target sensor. To make sure that there is no significant difference of the belt order, we conduct a few tests with both orderings. From the results of these tests, we do not notice any significant difference. When the belts are positioned so close together, they are expected to capture an equal magnitude of motion regardless of the ordering.

The effect of minor belt misplacement is especially important to measure when people are to use such equipment at home without the help of trained personnel. Additionally, the belts might just as well slip and change position as the subject moves or changes sleeping position in bed. We, therefore, conduct a number of preliminary experiments in that regard. To measure the effect of misplacement, we correctly fit the gold standard

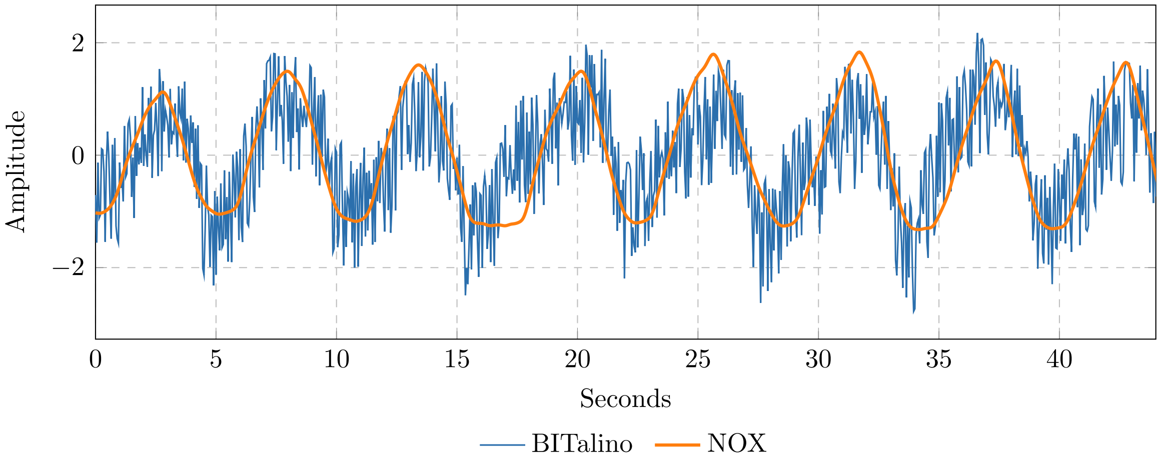


Figure 5.12: BITalino belt contaminated by higher frequency noise

with the target sensor placed about *5cm* off from its optimal position. We then move the target sensor steadily closer to its optimal position as time passes. More specifically, the first two minutes are measured with a *5cm* gap between the belts, the next two minutes with a *2.5cm* gap, and the last two minutes with (almost) no gap. This makes it possible to determine if and how much the signal quality improves as the belts approach their optimal positions. We conduct multiple experiments like this while alternating the direction of the misplacement (up or down).

The results of these tests show that *minor* misplacement of the belts does not matter too much regarding the quality of the produced signals. However, as the misplacement get more severe, the breath amplitudes are more affected. When the thoracic belt approaches the abdomen, the breath amplitudes increase, while they decrease as the belt approaches the top of the thorax (throat). Positioning the thoracic belt too close to the abdomen is, however, not desirable as the movement from the thorax and abdomen must be measured independently to estimate tidal volume accurately.

On a side note, we did encounter the oddity regarding BITalino flipping the signal across the y-axis when we repositioned the belts during these tests. This further suggests that the phenomenon is triggered by physical movement and does not only occur during sleep. Furthermore, we discovered yet another oddity during these tests. The signals from the BITalino belts were heavily contaminated by higher frequency noise (both the abdominal and thoracic belt). See Figure 5.12 for an example. The cause of this phenomenon is yet unknown, but it seems to be caused by some sort of interference. It did not change with any physical movement or readjustment of the belts. After conducting a frequency analysis of the signal, we find that the magnitude of the frequencies involved are evenly distributed across all bands above 0.5 Hz (which is where the respiratory component ends), i.e., 0.5–10 Hz, which suggests that it is *white noise*.

Shimmer Experiments

As described in Section 4.4.2, Shimmer uses an *impedance plethysmography* (IP) type sensor, which utilizes electrodes attached directly to the skin on the subject's chest and upper thigh. In contrast to, for example, a RIP type sensor, this technology is regarded

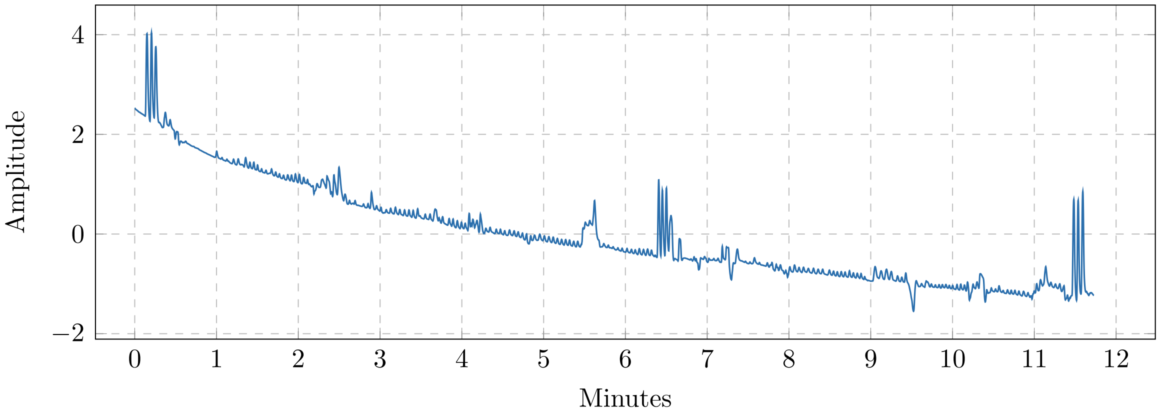


Figure 5.13: Overview of the first twelve minutes of a Shimmer capture

as being more affected by noise, while also suffering from baseline wander (Brouillette et al. 1987). Shimmer is, as the NOX, certified as medical grade equipment (Shimmer 2018a) and its general quality is, therefore, expected to be reasonably good. Our intention for conducting these preliminary tests with Shimmer is to uncover if any of these expected differences should be taken into consideration when designing the signal capture procedure. In addition, we must also assure that the addition of the electrodes does not affect the performance of the other sensors in any way. Having four respiratory effort belts in addition to three electrodes and the Shimmer device itself can be troublesome.

As seen in Figure 5.13, the baseline wander is present (as expected) and clearly visible in the signal. This is the first twelve minutes of a signal capture, and the spikes are prominent breaths (i.e., not noise or motion artifacts). As time passes, the baseline wander does stabilize a little, and instead of going just downwards, it starts to vary by going up and down like a low-frequency component. It does, however, take quite a few minutes (around 30 or so) before this starts to happen. Baseline wander with such a low frequency as this, is rather easily corrected (or at least improved) by fitting and subtracting a low-order polynomial from the signal.

The electrodes attached to the subject's chest are located at about the same level as the respiratory effort belts, although on the side of the chest under the arms. This means that the belts are strapped atop of the electrodes. The respiratory effort belts are, however, meant to be worn atop of clothing, and the extra electrodes did not affect the signal in any way in the tests we performed.

5.3 Signal Capture Procedure

With the results of the preliminary testing in mind, we design in this section a procedure to capture data from various external subjects. The design of this signal capture procedure defines, in other words, *how* we acquire data from the sensors. This includes, for example, how long each signal capture should be, what positions the subject should undertake, and what actions the subject may perform throughout the procedure. The primary goal is, in other words, to gather as many representative captures as needed to make the results as reliable as possible.

5.3.1 Setting

Body Position

The supine (back) and lateral decubitus (side) positions are the most common sleeping positions mentioned in the literature. The prone position (chest down) is less commonly mentioned, which might be due to certain position restrictions enforced by the equipment. In many studies, the subjects are either restricted to certain positions due to the equipment or have been instructed to sleep in a specific position the whole night for consistency. Our equipment also puts a restriction on the prone position as the NOX device itself is located on the subject's chest, which makes the prone position very uncomfortable. The relevant body positions for our signal capture procedure are, therefore, the *supine* (back) and *lateral decubitus* (side) positions. Many studies do not even differentiate between the left and right side, e.g., (George et al. 1988), as they are mostly the same. As such, we do not differentiate between the left and right side either.

In a study by Whyte et al. (1991), they conclude that RIP belts are unreliable when the subjects are allowed to change body position throughout the night. We have already seen this phenomenon from the preliminary tests, where the baseline breath amplitude differs significantly between different body positions; although, not visually noticeable for the RIP-belts in our tests. This further suggests that changes in body position should be included in the signal capture procedure.

Based on this information and our requirement for different body positions (Section 4.3.2), we include the *supine* (back) and *lateral decubitus* (side) positions in our signal capture procedure. By alternating between the supine and side positions during the test sessions, we further uncover the frequency and impact of the breath amplitude changes; not to forget any quality differences that might be present between the different positions.

Breathing Style

To make the signal captures more representative for sleep apnea monitoring, we require that both breathing stops as well as different breathing styles are included in the signal captures (Section 4.3.2). To simulate apneic events, the breathing stops should be at least ten seconds in duration. Real apneic events last from ten seconds up to multiple minutes, depending on the severity of the disorder. A sensor must, in other words, be able to detect breathing stops of at least ten seconds in duration to be usable. Longer gaps may, however, be split into multiple smaller gaps whenever the sensor is exhibiting false breaths. The duration of the simulated breathing stops does not need to be significantly longer than ten seconds, but a little longer may be beneficial to measure any inaccuracies. A duration of about *10–20 seconds* should be sufficient.

To better be able to measure the breath amplitude accuracy, we require (Section 4.3.2) that periods of both normal breathing, deep breathing, and shallow breathing are included in the captures. This results in three different levels of breath amplitudes (shallow, normal, deep) in addition to minor variations between each breath. How much deeper/shallower the breathing should be is hard to define definitively, and will, thus, be subjective and vary between different signal captures. Variation like this is good to better capture the general case. The inclusion of shallow breathing simulates hypopneic

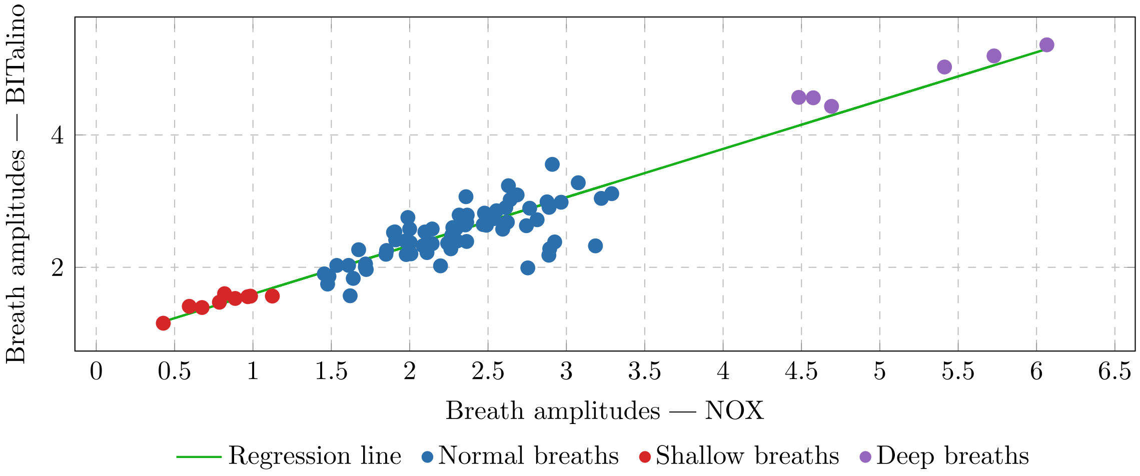


Figure 5.14: The effect of including both shallow, normal, and deep breaths

events, and should, therefore, last a minimum of ten seconds. With the same reasoning as for apneic events, a duration of 10–20 seconds should be sufficient. The duration of the periods with deep breathing does not need to be too long either, but it should differ from normal breathing. A duration of about 10–20 seconds should be sufficient for these periods as well.

The effect and importance of including three different breathing styles are illustrated in Figure 5.14. Because we are using a regression algorithm that is robust against outliers, the relationship is still expected to get captured if either one of the shallow or deep breathing periods happens to be very inaccurate.

5.3.2 Duration

The choice of how long the test sessions should be is a trade-off between quantity versus quality (or representability) of tests. As previously mentioned, signal capture sessions performed overnight when the subject is asleep is with no doubt the most representative situation for sleep apnea monitoring. These longer sessions, however, require significantly more work, both with respect to execution as well as subject recruitment, compared to shorter sessions that can be performed in the laboratory during wakefulness. A decent quantity of tests is essential to determine the frequency of some of the oddities discovered during the preliminary testing (and additional yet undiscovered ones), in addition to increasing the confidence and the generalizability of the signal quality measurements. Shorter tests performed during wakefulness are, in other words, preferred, as long as they are sufficiently representative (see Section 4.3.2).

In related literature, the duration of the signal captures varies between studies. Some perform overnight captures, for example, (Wu et al. 2009), (Whyte et al. 1991), and (Cantineau et al. 1992), while others perform shorter sessions captured during wakefulness, for example, (Retory et al. 2016), (Liu et al. 2013), and (Pennock 1990). Although worth mentioning is that not all of these studies focus specifically on sleep apnea. In a study by Cantineau et al. (1992), they found that the accuracy of RIP belts varies between wakefulness and different stages of sleep, with REM sleep showing the most inaccurate results. This suggests that the accuracy may decline over the course of the night as REM sleep becomes more prominent as the night progresses. In other words, if we only perform the test sessions during wakefulness, the results might be biased. However, the accuracy they describe in this study represents breath amplitude consistency with respect to an up-front calibration against a pneumotachograph. All of our sensors measure abdominal and thoracic movement/expansion, whereas a pneumotachograph measures airflow as inhaled through the mouth/nose. There has to the best of our knowledge not been conducted any studies showing that the difference between different respiratory effort sensors vary in relation to varying stages of sleep, and it might, therefore, not be too relevant for our case. As already seen from the preliminary tests, the BITalino sensor exhibits highly inconsistent breath amplitudes between body position changes. If this behavior is the norm for the sensors, then it suggests that a static up-front calibration proves merely ineffective for these sensors; at least when the subjects are allowed to change body position throughout the night.

To summarize the information we have gathered regarding the choice of signal capture duration:

- The preliminary tests indicate that:
  - A two minute initialization period is necessary.
  - The overall breath detection accuracy is captured during the first fifteen minutes.
  - No consistent breath amplitude accuracy degradation over time is present.
  - All sensor specific oddities are discoverable during shorter wakeful sessions.
  - Position changes have a massive impact on the breath amplitudes.
- The durations used by related studies vary from one minute to overnight captures.
- And the requirements we specified in Section 4.3.2 are:
  - A signal capture must include:
    - Breathing stops.
    - A period of shallow breathing.
    - A period of deep breathing.
    - Multiple sleeping positions.
  - The subject must be lying in bed.

The length of a signal capture must, in other words, be long enough to include (1) a two-minute initialization period, (2) two sleeping positions, (3) breathing stops, (4) a period of shallow breathing, and (5) a period of deep breathing. Additionally, one must remember that the breath amplitudes change considerably between different sleeping positions. If we are to compare the periods of different positions separately, they must be long enough to include enough data points to fit the linear regression model properly.

Based on this information, sessions lasting around *sixteen minutes* should be long enough to capture the overall signal quality of a sensor. Sessions of this length allow for at least one position change, while still being short enough to not become unbearably boring for the subjects. Since we only differentiate between two different body positions (supine and side), a signal capture of sixteen minutes gives about seven minutes of data from each position (excluding the first two minutes if necessary). Seven minutes for each position gives about 84–126 (breaths) data points, which should be sufficient to fit the linear regression model properly. During each body position period, there are four events of disrupted/alternate breathing, which means that there are at least three minutes from each position with normal breathing.

Regarding the initial baseline wander of the Shimmer sensor, the degree and behavior do stabilize after about 30 minutes (note that *stabilize* does not necessarily mean *improve* in this situation). Besides, the present baseline wander is easily improved by fitting and subtracting a low-order polynomial from the signal. Capturing the first sixteen minutes of the Shimmer signal is, therefore, not expected to affect the outcome of the results in any way.

5.3.3 The Signal Capture Procedure

Based on the discussions above, we arrive at the following signal capture procedure:

- Each signal capture is approximately sixteen minutes in length.
- At least seven consecutive minutes are captured from each body position (side and supine).
- Two breathing stops lasting 10–20 seconds, in each body position.
- One period of shallow breathing lasting 10–20 seconds, in each body position.
- One period of deep breathing lasting 10–20 seconds, in each body position.
- For the remaining duration, the subject breathes normally while lying still.

The subject is shown an example of what shallow and deep breathing means beforehand, but the performance is, nonetheless, subjective. The exact times and order of the periods of disrupted breathing and body positions are not strictly defined. However, for consistency, all subjects perform the following signal capture procedure in our experiments:

- Minute 1–9: Subject lies in the supine position.
  - Minute 3: Subject holds their breath for seventeen seconds.
  - Minute 4: Subject holds their breath for seventeen seconds.
  - Minute 5: Subject breathes shallowly for seventeen seconds.
  - Minute 6: Subject breathes deeply for seventeen seconds.
- Minute 10–16: Subject lies in the side position.
  - Minute 12: Subject holds their breath for seventeen seconds.
  - Minute 13: Subject holds their breath for seventeen seconds.
  - Minute 14: Subject breathes shallowly for seventeen seconds.
  - Minute 15: Subject breathes deeply for seventeen seconds.

5.4 Discussion and Conclusions

To summarize, we perform the following preprocessing steps of the raw signals:

1. Calculate the uncalibrated RIP_{sum} signal.
2. Remove baseline wander.
3. Integrate airflow signal to volume.
4. Resample the signals to 20 Hz.
5. Synchronize with the use of *cross-correlation*.
6. Standardize the signals.

A breath in the signal is the same as a *peak*, which we automatically score with the *findpeaks* function from MATLAB before we manually confirm and validate the detected peaks. The *start/end* of a peak is defined as the minimum value between two peaks, and the amplitude of a breath is calculated as the mean difference between its peak and start/end.

We have three breath detection accuracy metrics: *sensitivity*, *positive predictive value*, and *clean minute proportion*, along with one breath amplitude accuracy metric: *weighted absolute percentage error*. Sensitivity yields the proportion of correctly identified real breaths in relation to all real breaths, whereas PPV yields the proportion of real breaths in relation to all detected breaths. CMP yields the proportion of minutes in the signal that are 100% accurate in regards to breath detection. WAPE yields the mean amplitude error of the breaths in relation to the *baseline breath amplitude*. The baseline breath amplitude is defined as the mean of the breath amplitudes in the signal, and the expected amplitude of a breath is acquired through linear regression.

Chapter 6

Implementation

To speed up the process of the preprocessing and the calculation of the signal quality metrics, we describe in this chapter the implementation of several automatic scripts. These scripts involve first and foremost the following: (1) data format conversion, (2) preprocessing and synchronization, and (3) signal quality measurement. The file formats exported by the data acquisition software from the different platforms are not equal. While both Noxturnal and Consensys can export the data to a *comma-separated* (CSV) file format, the BITalino acquisition software by Gjøby (2016) supports only JavaScript Object Notation (JSON). As such, the file formats from the different platforms need to be unified. The intention behind the preprocessing and synchronization script is to prepare the signals for the quality evaluation according to the procedure defined in Chapter 5. At last, the quality measurement script accepts the input generated by the preprocessing script and calculates the signal quality metrics.

We begin in Section 6.1 by describing the system environment we use during the evaluation process. This includes the programming language of the scripts, the library versions, and the operating system. We continue in Section 6.2 and Section 6.3 by describing the file format conversion and RIP$_{\text{sum}}$ generation scripts. Next, we describe the preprocessing script in Section 6.4, before we present the quality measurement script in Section 6.5. Finally, we summarize and conclude the chapter in Section 6.6.

6.1 System Environment

We have decided to implement the scripts using Python, mainly because there are libraries available for Python that already contain most of the needed functions. The libraries we use include *pandas* (pandas 2018), *NumPy* (NumPy 2017), *SciPy* (SciPy 2018), and *scikit-learn* (scikit-learn 2017), which together, contain functions for CSV-parsing, resampling, interpolation, synchronization, integration, standardization, regression, and more, as well as a large number of utility functions. Additionally, we also use the library called *matplotlib* (matplotlib 2017) to plot the data in a chart. We have, however, not been able to find any decent implementations of peak detection for Python that meets our requirements. Most of the Python implementations we have seen do not support a definition of peak height independent of its y-value (Tournade 2015). As previously mentioned, the value of a peak of a breath may very well be negative because of noise and baseline wander, and should, therefore, be measured as the difference between its peak

Software	**Version**
Python	2.7.14
pandas	0.22.0
NumPy	1.14.0
SciPy	1.0.0
scikit-learn	0.19.1
matplotlib	2.1.2
MATLAB	R2016b
MacOS	10.13.3

Table 6.1: Software versions used during the quality evaluation

and its start/end values. Fortunately, the *findpeaks* function available in MATLAB (The Mathworks, Inc. 2016a) does partially support this definition through its *peak prominence* definition, which is explained further in Section 6.5.2. The functions from MATLAB can easily be called directly from Python as if it were a normal library, and so the use of MATLAB causes no additional problems. MATLAB must, nonetheless, be installed on the system, along with the MATLAB Python API which comes bundled as a part of the package. An overview of the software versions we are using during the signal quality evaluation is given in Table 6.1.

pandas (pandas 2018) is an open-source library which contains a number of data structures and tools specifically designed for data analysis. Functionality such as CSV-parsing, resampling, and interpolation are readily available. Another useful utility function is the *shift* function, which shifts the signal along its x-axis such that synchronization of signals becomes trivial.

NumPy (NumPy 2017) is the fundamental open-source library for scientific computing. It is most widely known as an n-dimensional array package, but it is loaded with other useful utility functions as well; especially with respect to vector/matrix arithmetic.

SciPy (SciPy 2018) is an open-source library which contains a wide range of different scientific functionality, such as signal processing capabilities. *SciPy* contains functionality for integration, cross-correlation, and frequency filtering.

scikit-learn (scikit-learn 2017) is an open-source library for data mining, machine learning, and data analysis, which is built on *NumPy*, *SciPy*, and *matplotlib*. In our implementation, we make use of its *TheilSen* linear regressor.

matplotlib (matplotlib 2017) is an open-source 2D plotting library for Python, which is designed to be easy and intuitive to use. It supports a wide range of charts, and we use it for plotting the signals themselves in a line chart, as well as the breath amplitudes in a scatterplot. The charts also have support for interaction, which we use to manually validate and correct the breath detection.

```
{
  "type": "data",
  "id": 0,
  "time": "13:32:44:235",
  "data": [
    {
      "id": 0,
      "value": 396
    },
    {
      "id": 1,
      "value": 364
    }
  ]
}
```

Listing 6.1: Example of JSON output from the BITalino acquisition application

MATLAB (The Mathworks, Inc. 2016b), also known as Matrix Laboratory, is a commercial mathematical platform by The Mathworks, Inc. (2018) which includes its own scripting language. MATLAB contains a wide range of functionality for machine learning, data mining, signal processing, and so forth. Although MATLAB is an independent platform, it can be interfaced and used from a wide range of other programming languages, for example, Python. We use it primarily for its *findpeaks* function (see Section 6.5.2).

6.2 Data Format Conversion

The data acquisition software for BITalino is storing the captured data in JSON format. Every sample is stored as its own JSON object on the form shown in Listing 6.1, with one such JSON object per line in the file. In other words, the file as a whole is not valid JSON, but each line separately is. To unify the file formats amongst the different platforms, we convert this JSON format to a *comma-separated* (CSV) file format. This conversion is straightforward, and the code of the python script we use for this conversion is shown in Listing 6.2. This script accepts two command line arguments, the first is the input file name of the file containing the JSON, and the second is an optional sensor index. The intention behind this optional sensor index is to be able to separate the data from different sensors to different CSV-files, if desired, such as the abdominal and thoracic signals. This script writes the generated CSV-file to *standard output*, which can then be redirected to a file. To give an example of how to execute this script:

```
$ python csv-converter.py bitalino.txt 0 > bitalino-thorax.csv
```

With this command, the sensor index of the thorax signal is *0*, which is then extracted from *bitalino.txt* and stored in the new CSV-file named *bitalino-thorax.csv*.

6.3 Generation of the RIP$_{\text{sum}}$ Signal

The generation of the RIP$_{\text{sum}}$ signal is, as specified in Chapter 5, a part of the preprocessing. However, as a separation of concerns, this logic is separated into its own script. The script is in its essence very simple, as all it needs to do is to read the supplied input CSV-files, add the values together, and then write out the result as a new CSV-file.

```
import json, sys

# Read command line arguments
try:
    INPUT_FILE = sys.argv[1]
    SENSOR_INDEX = int(sys.argv[2]) if (len(sys.argv) > 2) else None
except:
    print("Usage: %s <INPUT_FILE> [<SENSOR_INDEX>]" % sys.argv[0])
    exit(1)

infile = open(INPUT_FILE, 'r')
for line in infile:
    try:
        json_data = json.loads(line)
        if json_data['type'] != 'data': continue

        if SENSOR_INDEX >= 0:  # either only one sensor
            values = str(json_data['data'][SENSOR_INDEX]['value'])
        else: # or all sensors separated by \t
            values = '\t'.join(map(lambda e: str(e['value']), json_data['data']))

        print(json_data['time'] + '\t' + values)

    except: continue
```

Listing 6.2: JSON to CSV converter

Unfortunately, it needs to be slightly more complex than this. The reason is that the abdominal and thoracic signals from the NOX sensor are not synchronized, and each sample is captured two milliseconds apart from each other. Because of this fact, the script also needs to synchronize the input signals. Additionally, one or both of the signals from BITalino are sometimes flipped across its y-axis. When only one of the signals is flipped, it needs to be corrected before the values can be arithmetically combined. The code of the script can be seen in Listing 6.3, with the argument parsing omitted for clarity, and it can be executed with the following command:

```
$ python csv-combiner.py --fs=20 --file=bitalino-thorax.csv,f
  --file=bitalino-abdomen.csv > bitalino-ripsum.csv
```

In this example, the thoracic and abdominal signals from BITalino are resampled to 20 Hz, synchronized, and then arithmetically combined to produce the output. Moreover, the thoracic signal is flipped across its y-axis before the signals are combined. The `--fs=20` argument specifies that the sampling rate should be set to 20 Hz (sampling rate is required for synchronization), and the `,f` suffix of the thoracic signal's file name specifies that this signal should be flipped. A *nosync* and a *delay* argument are also supported. This script is mostly a stripped down version of the preprocessing script, and so the details of the synchronization and arguments are described further in Section 6.4.

6.4 Preprocessing

To preprocess the signal data, we utilize the chosen *pandas, NumPy* and *SciPy* libraries. Together, these libraries already contain most of the functions needed for the preprocessing, including resampling, interpolation, synchronization, integration, and standardization. The main part of the preprocessing script can be seen in Listing 6.4, with the argument parsing omitted for clarity. The required arguments for this script are the *sampling*

```python
# Read CSV-files
signal1 = pandas.read_csv(files[0]['name'], sep="\t", header=None, index_col=0,
    parse_dates=[0], date_parser=parseTimestamp)
signal2 = pandas.read_csv(files[1]['name'], sep="\t", header=None, index_col=0,
    parse_dates=[0], date_parser=parseTimestamp)

# Resample and interpolate
fs_interval = str(1000 // args['fs']) + 'ms' # e.g., 20 Hz == one sample every 50ms
signal1 = signal1.resample(fs_interval).mean()
signal2 = signal2.resample(fs_interval).mean()

signal1 = signal1.interpolate(method='quadratic')
signal2 = signal2.interpolate(method='quadratic')

# Flip
signal1[1] = applyFileOptions(signal1[1].values, files[0])
signal2[1] = applyFileOptions(signal2[1].values, files[1])

# Synchronize
if 'nosync' not in args:
    delay = su.findDelay(signal1[1], signal2[1])
    signal1 = signal1.shift(delay)
if 'delay' in args: signal1 = signal1.shift(args['delay'])

# Drop NaNs
signal1 = signal1.dropna()
signal2 = signal2.dropna()

# Equalize lengths
signal1 = su.cutLengthOf(signal1, to=signal2)
signal2 = su.cutLengthOf(signal2, to=signal1)

# Arithmetically add their y-values
combined = signal1 + signal2

# Write new csv-file
combined.to_csv(sys.stdout, sep='\t', header=False, date_format='%H:%M:%S:%f')
```

Listing 6.3: Script to combine CSV-files to generate the RIP$_{\text{sum}}$ signal

rate and the two input CSV signal files. Additionally, the script also accepts optional arguments for *nosync*, *delay*, *start/end* location, and specific options for the different input files. The *nosync* argument disables the automatic cross-correlation synchronization. The optional *delay* argument is meant as a fine-tuning argument for the synchronization and is applied after the automatic cross-correlation method, regardless of the automatic method being disabled or enabled. The delay argument is specified as an integer (can be negative) in the number of samples the first signal should additionally be delayed. The *start* and *end* arguments specify the part of the signal that should be included in the output. In other words, only the samples between *start* and *end* are included in the output. The input file specific options are specified as a part of the file name, separated by commas. The syntax is given as follows: `--file=<NAME>,[i],[dt=<ORDER>],[h=<LOWCUT>],[f]`. The *i* option indicates that the signal should be *integrated* from *flow* to *volume*, and *dt* specifies that the signal should be *detrended* with a polynomial of the given order. The *h* option specifies that the signal should be filtered with a high-pass filter with the given low-cut frequency, and *f* flips the signal across its y-axis. The output of this script is a new CSV-file where the input files are combined, preprocessed, and synchronized. Again, the output is written to *standard output*, which can be redirected to a new file if desired. Additionally, this script plots the processed data in a chart when the optional argument *show* is supplied. This way, the synchronization and preprocessing can be confirmed and verified visually, and altered if desired.

To give an example of how to run this script:

```
$ python preprocess.py --fs=20 --delay=11 --start=2000 --end=10000 --show
    --file=bitalino-abdomen.csv,i,h=0.1,f --file=nox-abdomen.csv > preprocessed.csv
```

For this example, most of the arguments are specified. The `--fs=20` argument specifies that the sampling rate should be set to 20 Hz, and the `--delay=11` argument specifies that the signal should be delayed an additional *11* samples. The `--start=2000` and `--end=10000` arguments specify that the signal in the period 2000–10000 (in the number of samples) should be extracted, while the rest of the signal discarded. For the file specific options, the `--file=bitalino-abdomen.csv,i,h=0.1,f` argument specifies that the BITalino signal should be *integrated* to volume, filtered with a high-pass filter with a low-cut frequency of 0.1 Hz, and flipped. The second input file contains no custom options. The result of this preprocessing is a new CSV-file stored as *preprocessed.csv*, with one column per input signal.

6.4.1 Resampling

Resampling is already available in the *pandas* library through the function *resample*. This function, however, accepts the sampling rate as an *interval* rather than a frequency in Hertz. For example, an interval of *50ms* means that there should be one sample every 50 milliseconds. To convert between a sampling rate specified in Hertz to an interval, the following formula is applied: $interval = \frac{1000}{Hz}$. As one may notice in the script, both input signals are resampled regardless of their original sampling rates. In other words, the NOX signal is resampled even though it is already sampled at 20 Hz. The effect of this is that the timestamp of each sample is adjusted, whereas the samples themselves remain untouched. In other words, the signal is shifted along its x-axis such that the sample's timestamp has a millisecond value which is divisible by the specified sampling interval (the interval is 50ms for 20 Hz). This makes it possible to align the timestamps

```python
# Read CSV-files
signal1 = pandas.read_csv(files[0]['name'], sep="\t", header=None, index_col=0,
    parse_dates=[0], date_parser=parseTimestamp)
signal2 = pandas.read_csv(files[1]['name'], sep="\t", header=None, index_col=0,
    parse_dates=[0], date_parser=parseTimestamp)

# Resample
fs_interval = str(1000 // args['fs']) + 'ms' # e.g. 20 Hz == one sample every 50ms
signal1 = signal1.resample(fs_interval).mean()
signal2 = signal2.resample(fs_interval).mean()

# Interpolate
signal1 = signal1.interpolate(method='quadratic')
signal2 = signal2.interpolate(method='quadratic')

# Equalize lengths
signal2 = su.cutLengthOf(signal2, to=signal1)
signal1 = su.cutLengthOf(signal1, to=signal2)

# Extract part if specified
start = args['start'] if 'start' in args else 0
end = args['end'] if 'end' in args else len(signal1[1])
signal1 = signal1.iloc[start:end]
signal2 = signal2.iloc[start:end]

# Filter, detrend, integrate, flip
signal1[1] = applyFileOptions(signal1[1].values, files[0], args['fs'])
signal2[1] = applyFileOptions(signal2[1], files[1], args['fs'])

# Standardize
signal1 = su.standardize(signal1)
signal2 = su.standardize(signal2)

# Synchronize
if 'nosync' not in args:
    delay = su.findDelay(signal1[1], signal2[1])
    signal1 = signal1.shift(delay)
if 'delay' in args: signal1 = signal1.shift(args['delay'])

# Drop NaNs
signal1 = signal1.dropna()
signal2 = signal2.dropna()

# Equalize lengths
signal1 = su.cutLengthOf(signal1, to=signal2)
signal2 = su.cutLengthOf(signal2, to=signal1)

preprocessed = signal1 * 1 # make copy
preprocessed[2] = signal2[1] # add column
preprocessed.to_csv(sys.stdout, sep='\t', header=False, date_format='%H:%M:%S:%f')

if 'show' in args:
    pyplot.plot(signal1.index, signal1[1], label=files[0]['name'])
    pyplot.plot(signal2.index, signal2[1], label=files[1]['name'])
    pyplot.legend()
    pyplot.show()
```

Listing 6.4: The preprocessing script

```
def standardize(signal):
    return (signal - signal.mean()) / signal.std()
```

Listing 6.5: Function to standardize the signal

```
def findDelay(a, b):
    return (len(b) - 1) - np.argmax(signal.correlate(a, b))
```

Listing 6.6: Find delay function using cross-correlation

of the different signals perfectly. We resample by taking the *mean* value for samples that end up between existing samples. The mean might, however, not be available for a few samples (e.g., at the signal boundaries), and as such, the missing samples are interpolated using *quadratic interpolation*.

6.4.2 Standardization

To standardize the signals, we follow the formula as specified in Chapter 5. The function we use in the script (Listing 6.4), can be seen in Listing 6.5, where the *mean* is subtracted from every sample, and then each sample is divided the *standard deviation*. The use of the *DataFrame* type supplied by the *pandas* library makes this very trivial to do because of its built-in utility functions and support for vector arithmetic.

6.4.3 Synchronization

Synchronization with the use of cross-correlation is already partially implemented as a part of the *SciPy* library, making it rather trivial to adjust to our context. The *correlate* function from *SciPy* returns an array of correlation values for all possible alignments of the signals. After the formula specified in Section 3.2.1, the index of the max value in this array reflects the correct synchronization point. We have wrapped the function from *SciPy* in a *findDelay* function (see Listing 6.6) which returns the delay between the signals as an integer. The first signal is shifted along its x-axis according to this delay (see Listing 6.4) to complete the synchronization.

6.4.4 Integratation

For the integration, we decided in Chapter 5 to use the *Trapezoidal Rule*. Fortunately, a cumulative version of this algorithm is already available in the *SciPy* library. The function is called *cumtrapz* and is located in the *integrate* module in the library. As we are cumulatively integrating from *flow* to *volume*, the integrated result contains one sample for every *interval* in the original signal. This means that the integrated result has one less sample compared to the original signal. To correct for this, we simply prepend a sample with the value *0* to the integrated signal. The integration is performed with the following statement: `signal = integrate.cumtrapz(signal, initial=0)`.

```python
def detrend(sig, order=7):
    coeff = np.polyfit(range(len(sig)), sig, order)
    model = np.poly1d(coeff)
    polynomial = [model(x) for x in xrange(len(sig))]

    return sig - polynomial
```

Listing 6.7: Function to detrend the signal

```python
def highpassFilter(sig, lowcut, fs, order=5):
    lowcut /= fs / 2 # Nyquist
    b, a = signal.butter(order, lowcut, btype='highpass')
    return signal.lfilter(b, a, sig)
```

Listing 6.8: The high-pass filter function

6.4.5 Detrending

To fit a polynomial to the signal, we utilize the *polyfit* function provided as a part of the *NumPy* library. This function uses the *least-squares* algorithm to fit the polynomial and then returns the coefficients in order of decreasing power. To compute the actual points of the polynomial using the x-values of the signal, we utilize the helper function *poly1d*. This function converts the mathematical polynomial into a runnable python function. The complete function we use for detrending the signal can be seen in Listing 6.7.

6.4.6 High-pass Filtering

We use a *Butterworth filter* for the high-pass filtering of the signals (Wikipedia 2018a). This filter is available as a part of the *SciPy* library through two functions, *butter* and *lfilter*. The *butter* function constructs the filter itself with the specified parameters, whereas the *lfilter* function performs the actual filtering of the signal using the constructed filter. The complete function can be seen in Listing 6.8.

6.5 Signal Quality Measurement

The quality measurement script (Listing 6.9) accepts the CSV-file generated by the preprocessing script as input, along with its sampling rate. In other words, it accepts a CSV-file with three columns, where the first column is the x-axis as timestamps, the second column is the target signal, and the third column is the reference/gold standard signal. In addition, optional arguments may also be provided, such as *validate*, *show*, *minAmplitude*, *minDuration*, and *maxDuration*. When the *validate* argument is provided, a chart of the two input signals is presented to the user, who may manually validate and correct the detected breaths. The *show* argument specifies that the result of the quality evaluation (detected breaths and breath amplitude relationship) shall be plotted visually in a chart. *minAmplitude*, *minDuration*, and *maxDuration* specify the minimum amplitude, minimum duration, and maximum duration of a breath, respectively. These three arguments have the values specified in the design as default values (i.e., 10%, 0.6s, and 12s, respectively). The output of this script is the four metrics *WAPE* (as the mean

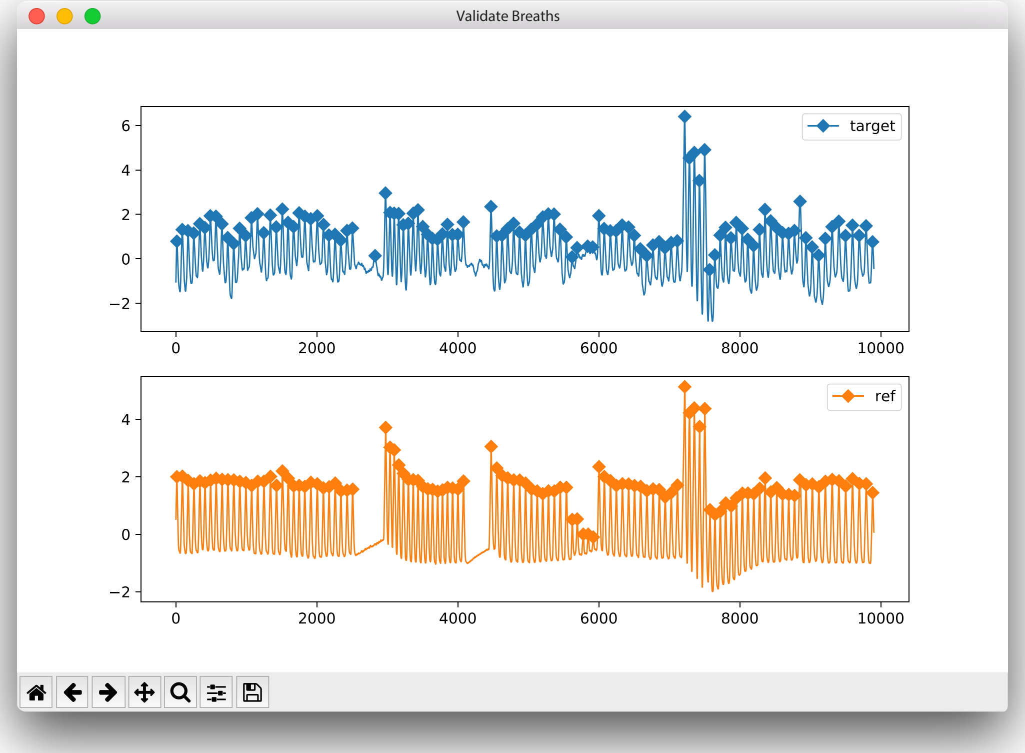


Figure 6.1: Manually validate and correct breaths

breath amplitude error), *sensitivity*, *positive predictive value*, and *clean minute proportion*.

To give an example of how to execute this script:

```
$ python measure.py --file=processed.csv --fs=20 --show --validate
```

The output may look like the following:

```
Mean breath amplitude error: 13.46% +- 8.48%
Breath sensitivity: 98.68%
Breath positive predictive value: 97.69%
Breath clean minute proportion: 0.00%
```

The dialog shown in Figure 6.1 is presented when the *validate* argument is provided. In this window, the user may zoom and scroll, and toggle breaths on/off by double-clicking in any of the two charts. When the breaths are validated, the user closes the window to continue execution of the script.

6.5.1 Breath Amplitude Approximation

The function we use to approximate the mean breath amplitude of a signal (by the logic described in Chapter 5), can be seen in Listing 6.10. As described, the signal is split into n windows with the given *width*, and then the *mean point-to-point* value of these windows is calculated. The *point-to-point* value is the difference between the maximum

```python
# Read CSV-file
signal = pandas.read_csv(args['filename'], sep="\t", decimal=".", header=None,
    index_col=0, parse_dates=[0], date_parser=parseTimestamp)

# Extract columns
targetY = signal[1].values
refY = signal[2].values

# Approximate mean breath amplitude with 4 sec window, and take the minAmplitude % as min threshold
minTargetBreathAmp = resp.approxMeanBreathAmp(targetY, fs * 4) * minAmplitude
minRefBreathAmp = resp.approxMeanBreathAmp(refY, fs * 4) * minAmplitude

# Detect breaths
targetPeaks = resp.detectPeaks(targetY, minTargetBreathAmp, minDuration)
refPeaks = resp.detectPeaks(refY, minRefBreathAmp, minDuration)

if 'validate' in args:
    targetPeaks, refPeaks = manuallyValidatePeaks(targetPeaks, targetY, refPeaks, refY)

targetBreaths = resp.constructBreaths(targetPeaks, targetY, minTargetBreathAmp, maxDuration)
refBreaths = resp.constructBreaths(refPeaks, refY, minRefBreathAmp, maxDuration)

# Match breaths
matchedTargetBreaths, matchedRefBreaths = resp.matchBreaths(targetBreaths, refBreaths)

if not matchedTargetBreaths.size:
    print("No breaths matched... exiting")
    exit(0)

# Get breath amplitudes of matched breaths
targetBreathAmplitudes = np.array([b.amplitude() for b in matchedTargetBreaths])
refBreathAmplitudes = np.array([b.amplitude() for b in matchedRefBreaths])

# Make regression line of breath amplitudes
regressor = TheilSenRegressor()
regressor.fit(refBreathAmplitudes.reshape(-1, 1), targetBreathAmplitudes)
regressionLine = regressor.predict(refBreathAmplitudes.reshape(-1, 1))

print("Mean breath amplitude error: %.2f%%, %.2f%%" %
    metrics.wape(regressionLine, targetBreathAmplitudes))
print("Breath sensitivity: %.2f%%" %
    resp.breathSensitivity(matchedTargetBreaths, refBreaths))
print("Breath positive predictive value: %.2f%%" %
    resp.breathPosPredValue(matchedTargetBreaths, targetBreaths))
print("Breath clean minute proportion: %.2f%%" %
    resp.cleanMinuteProportion(matchedTargetBreaths, targetBreaths, matchedRefBreaths, refBreaths, fs=fs))

if 'show' in args:
    targetMarks = [b.peakIndex for b in matchedTargetBreaths]
    refMarks = [b.peakIndex for b in refBreaths]
    pyplot.plot(signal.index, targetY, '-D', markevery=targetMarks, label="target")
    pyplot.plot(signal.index, refY, '-D', markevery=refMarks, label="ref")
    pyplot.legend()

    pyplot.figure()
    pyplot.scatter(refBreathAmplitudes, targetBreathAmplitudes)
    pyplot.plot(refBreathAmplitudes, regressionLine, 'm', label='regression line')
    pyplot.legend()
    pyplot.show()
```

Listing 6.9: The quality measurement script

```python
def approxMeanBreathAmp(signal, windowWidth):
    windows = np.array_split(signal, len(signal) / windowWidth)
    theSum = reduce((lambda s, w: s + w.ptp()), windows, 0)

    return theSum / len(windows)
```

Listing 6.10: Function to approximate the mean breath amplitude

and minimum value of a window, which is acquired through the *ptp* utility function from *NumPy*. Notice that the `len(signal) / windowWidth` expression may not result in an integer. The *array_split* function handles this behavior automatically, i.e., the last window may contain fewer samples compared the other windows.

6.5.2 Breath Detection

We have decided to use the *findpeaks* function from MATLAB for the automatic breath detection (The Mathworks, Inc. 2016a). This function is the only peak detection implementation we have found that is usable from Python, which partially supports a definition of y-value independent peak height. This is supported through a concept MATLAB defines as *peak prominence*:

> The prominence of a peak measures how much the peak stands out due to its intrinsic height and its location relative to other peaks. A low isolated peak can be more prominent than one that is higher but is an otherwise unremarkable member of a tall range.
>
> To measure the prominence of a peak:
>
> 1. Place a marker on the peak.
> 2. Extend a horizontal line from the peak to the left and right until the line does one of the following; it
>   - crosses the signal because there is a higher peak, or
>   - it reaches the left or right end of the signal.
> 3. Find the minimum of the signal in each of the two intervals defined in Step 2. This point is either a valley or one of the signal endpoints.
> 4. The higher of the two interval minima specifies the reference level. The height of the peak above this level is its prominence.

Following this definition of peak prominence, the determined breath amplitudes are wrong (relative to our definition) for all breaths that have a higher peak value than its immediate neighbors. However, since we are only using prominence to filter out peaks that are too small, the definition is correct for the majority of the concerned peaks. The only case where the filtering fails is when a small peak (with less than 10% of mean peak amplitude) has a y-value higher than its immediate neighbors. This is, however, expected to happen very rarely and can be significantly improved further by applying a high-pass filter to the signal. In other words, the use of prominence for filtering smaller peaks is very well adequate in our situation.

```python
def detectPeaks(signal, minAmp, minDuration):
    data = matlab.double(signal.tolist())

    _, locs = engine.findpeaks(data,'MinPeakProminence', float(minAmp),
        'MinPeakWidth', int(minDuration / 2), 'WidthReference', 'halfprom', nargout=2)
    return np.array(locs._data.tolist()).astype(int) - 1 # matlab uses 1-based indexing
```

Listing 6.11: Function to detect peaks using the *findpeaks* function from MATLAB

```python
def constructBreaths(peaks, signal, maxDuration):
    breaths = []
    maxPeakDistance = maxDuration // 2

    for i in xrange(len(peaks)):
        currPeak = peaks[i]
        # either index of prev breath, or start of signal
        prevPeak = peaks[i - 1] if (i > 0) else 0
        # either index of next breath, or end of signal
        nextPeak = peaks[i + 1] if (i < len(breaths) - 1) else len(signal)

        # cap interval to maxPeakDistance
        prevPeak = max(prevPeak, currPeak - maxPeakDistance)
        nextPeak = min(nextPeak, currPeak + maxPeakDistance)

        breathStart = np.argmin(signal[prevPeak:currPeak + 1]) + prevPeak
        breathEnd = np.argmin(signal[currPeak:nextPeak + 1]) + currPeak

        breaths.append(Breath(signal, breathStart, breathEnd, currPeak))

    return np.array(breaths)
```

Listing 6.12: Function to construct the breaths based on the detected peaks

In addition to using prominence to filtering out peaks based on their amplitude, we use prominence to filter out breaths that are too short. *findpeaks* supports this functionality based on the width of the peak measured at half its prominence. Since we defined the duration of a breath as the time distance between its *start* and *end*, the measurement at half the prominence should be approximately half the breath duration.

We must, however, emphasize that the accuracy of the automatic breath detection is not the primary focus of this thesis. We manually validate and correct the detected breaths anyway, and the automatic detection is only meant as a means to speed up the process.

The function we use to detect the peaks of the signals can be seen in Listing 6.11. This function converts the provided signal to a MATLAB compliant data structure before it runs the *findpeaks* function with the given parameters. As mentioned above, we specify the minimum prominence of a breath to be the minimum amplitude, and the minimum width to be half the minimum duration. After the peaks returned by *findpeaks* are manually validated (if the *validate* argument is specified), the breaths are constructed based on these peaks. The code for this can be seen in Listing 6.12, where the *start/end* of each breath is derived, and then stored in a *Breath* object together with its peak. One may also notice how we cap the interval from the peaks in relation to the max duration of a breath.

```
def matchBreaths(targetBreaths, refBreaths):
    targetMatch = []
    refMatch = []

    for i in xrange(len(refBreaths)):
        match = np.array(filter(lambda b:
            refBreaths[i].startIndex < b.peakIndex < refBreaths[i].endIndex, targetBreaths))
        if (len(match) > 0):
            matched = np.argmax([e.peak for e in match]) # if multiple breaths match, pick highest breath
            targetMatch.append(match[matched])
            refMatch.append(refBreaths[i])

    return np.array(targetMatch), np.array(refMatch)
```

Listing 6.13: Function to match breaths between signals

6.5.3 Matching Breaths

As specified in Chapter 5, a breath of the target signal is matched if its *peak* is between the *start* and *end* of a breath in the reference signal. However, there may be the case when multiple breaths seem to be matched with the same breath of the reference signal. To avoid this behavior, we match only the breath with the highest peak in such cases. The function we use to match breaths is shown in Listing 6.13.

6.5.4 Breath Amplitude Regression

The breath amplitude regression is (Listing 6.9) performed as specified in Chapter 5. The *TheilSen* regression algorithm is already implemented and readily available from the *scikit-learn* library, which we utilize. We fit the model by using the breath amplitudes from the reference signal (gold standard) as the x-axis, and the breath amplitudes from the target signal as the y-axis. Next, we generate the regression line by predicting the y-values from the breath amplitudes of the reference signal (i.e., predicting y from x).

6.5.5 Signal Quality Metrics

All the signal quality metrics are implemented after the formulas given in Chapter 5 and can be seen in Listing 6.14. For the *WAPE* metric, we utilize the vector arithmetic functionality from *NumPy* to simplify the implementation and improve the readability. The formula for *sensitivity* and *positive predictive value* are essentially the same, but we have, however, separated their implementations into distinct functions for readability. The *clean minute proportion* implementation is a little more complex compared to the other metrics. Each of the different breath arrays (matched/all for both target and reference) is first mapped to a list containing the minute each breath is contained within. For example, if a breath is contained in minute *3*, its value is *2* in this list (*0-indexed*). Through *NumPy's bincount* function, we create one *bin* for each minute, which contains the number of breaths contained within the minute. For example, *bin 0* contains the number of breaths contained in minute *0*. In other words, the result of *bincount* could look like [5, 11, 9, 12, ...], which means that minute *0* contains *5* breaths, minute *1* contains *11* breaths, and so on. The difference between the number of breaths per bin (minute) in the *allTarget* and *matchedTarget* arrays is the number of false breaths per minute. Likewise, the difference between the number of breaths per bin in the *allRef*

```python
def wape(predicted, actual):
    err = abs(np.array(predicted) - np.array(actual)) / np.array(actual).mean() * 100.0
    return err.mean(), err.std()

def breathSensitivity(correctlyDetectedBreaths, allRealBreaths):
    return float(len(correctlyDetectedBreaths)) / len(allRealBreaths) * 100.0

def breathPosPredValue(correctlyDetectedBreaths, allDetectedBreaths):
    return float(len(correctlyDetectedBreaths)) / len(allDetectedBreaths) * 100.0

def cleanMinuteProportion(matchedTargetBreaths, targetBreaths, matchedRefBreaths,
    refBreaths, falseBreathTolerance=0, missedBreathTolerance=0, fs=20.0):
    samplesPerMin = int(fs * 60)
    totalMinutes = (max(targetBreaths[-1].peakIndex, refBreaths[-1].peakIndex) // samplesPerMin) + 1

    allTarget = np.bincount([b.peakIndex // samplesPerMin for b in targetBreaths],
        minlength=totalMinutes)
    matchedTarget = np.bincount([b.peakIndex // samplesPerMin for b in matchedTargetBreaths],
        minlength=totalMinutes)
    allRef = np.bincount([b.peakIndex // samplesPerMin for b in refBreaths],
        minlength=totalMinutes)
    matchedRef = np.bincount([b.peakIndex // samplesPerMin for b in matchedRefBreaths],
        minlength=totalMinutes)

    falseBreaths = [m > falseBreathTolerance for m in (allTarget - matchedTarget)]
    missedBreaths = [m > missedBreathTolerance for m in (allRef - matchedRef)]
    dirtyMinutes = np.logical_or(falseBreaths, missedBreaths)
    dirtyMinutes = dirtyMinutes[dirtyMinutes]

    return (1 - len(dirtyMinutes) / float(totalMinutes)) * 100.0
```

Listing 6.14: Metric implementations

and *matchedRef* arrays is the number of missed breaths per minute. For these, we count the number of bins (minutes) that violate the given thresholds. These thresholds are, however, by default set to *0*, which are the thresholds we use during the evaluation. At first glimpse, it may look like the *matchedRef* array is unnecessary to calculate missed breaths. After all, the difference between the breaths in *matchedTarget* and *allRef* also yields the number of missed breaths. However, as we are determining which minute each breath is contained within based on their *peaks*, there may be edge cases at the boundaries of each minute. The breath from the target signal *may* be matched with a breath contained in adjacent minutes. To avoid such situations, we use the *matchedRef* array instead.

6.6 Discussion and Conclusions

To summarize, we have implemented a total of four Python scripts, where each of these scripts is used at a different level in the quality measurement procedure. Such a procedure may involve the following steps: convert all the data to CSV-files using the *csv-converter.py* script. Next, the data from the thorax and abdomen may optionally be combined to create the RIP$_{\text{sum}}$ signal using the *csv-combiner.py* script, before the signals are preprocessed by the *preprocess.py* script. After the output from the preprocessing script is acquired, it is used as input to the *measure.py* script to evaluate the quality of the signals.

A complete procedure may be executed as follows:

1.
   ```
   $ python csv-converter.py bitalino.txt 0 > bitalino-thorax.csv
   $ python csv-converter.py bitalino.txt 1 > bitalino-abdomen.csv
   ```

2.
   ```
   $ python csv-combiner.py --fs=20 --file=bitalino-thorax.csv,f
       --file=bitalino-abdomen.csv > bitalino-ripsum.csv
   $ python csv-combiner.py --fs=20 --file=nox-thorax.csv
       --file=nox-abdomen.csv > nox-ripsum.csv
   ```

3.
   ```
   $ python preprocess.py --fs=20 --start=2000 --end=10000
     --file=bitalino-ripsum.csv,i,h=0.1 --file=nox-ripsum.csv > preprocessed.csv
   ```

4.
   ```
   $ python measure.py --file=processed.csv --fs=20
   Mean breath amplitude error: 13.46% +- 8.48%
   Breath sensitivity: 98.68%
   Breath positive predictive value: 97.69%
   Breath clean minute proportion: 0.00%
   ```

Chapter 7

Evaluation

This chapter presents the evaluation of the experiment results for the different platforms, a platform comparison, and a review of how well suited the metrics are for these kinds of data. We begin in Section 7.1 by presenting statistics and information about the recruited subjects, and continue in Section 7.2 with a few examples of the signals recorded from these subjects. Next, we evaluate the results of the BITalino and Shimmer platforms separately in Section 7.3 and Section 7.4, before we present a comparison of the best signals from these platforms in Section 7.5. We discuss and evaluate how well suited the metrics are for these kinds of data in Section 7.6, and present a comparison with related work in Section 7.7. Next, we present the results of two additional sensors, RespiBAN and FLOW, in Section 7.8, and finally, we conclude the chapter in Section 7.9. As a side note, keep in mind that this chapter is only concerned with BITalino and Shimmer until Section 7.8.

7.1 Subjects

We recruited a total of twelve healthy subjects, five females and seven males, with a mean age of 37 ± 15 years and a BMI of 27 ± 5. Unless otherwise stated, we refer to these subjects as *Subject 1–12* sorted by their BMI, where *Subject 1* has the lowest BMI, and *Subject 12* has the highest BMI.

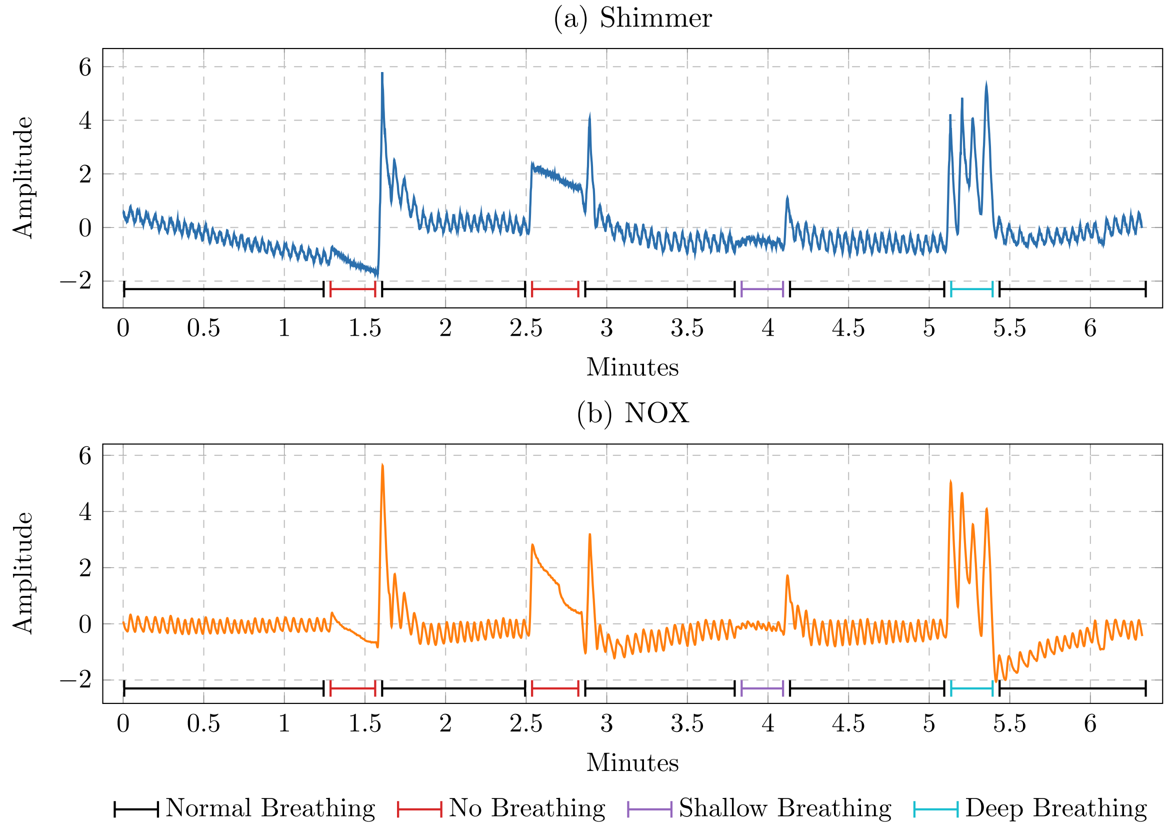
While most of the subjects were able to follow the signal capture procedure correctly, not everyone was able to hold their breath for the complete duration of a period. Instead of cutting the periods short, these subjects took one or two smaller breaths during the period. Many subjects stated that the hardest action to perform was the shallow breathing, but despite this fact, most subjects were still able to perform it correctly. Please note that the abdominal BITalino belt did not fit *Subject 1* very well. After tightening the belt as much as possible, it was still slightly more loose than what is optimal. We elaborate more on the impact of this below, but summarized, it did not affect the breath detection of the belt at all, but it did affect the breath amplitude accuracy quite significantly. For all the other subjects, the equipment fit perfectly.

7.2 Signal Capture Excerpts

To get a first impression of exactly what we measure the quality of, we present in this section a few examples of signal captures from the subjects. As presented in Chapter 5, the signal capture procedure consists of four actions per body position, two periods of no breathing, followed by a period of shallow breathing, followed by a period of deep breathing. Each of these periods lasts for seventeen seconds and are separated by slightly longer periods of normal breathing. Exactly when these periods occur during the signal captures are annotated in the examples shown below. A common feature visible in these examples is a deeper breath at the end of the periods of disrupted breathing, as the subjects are gasping for air. One may notice that not all the subjects were able to hold their breath for the complete duration of a period, and for some subjects, the waveform, therefore, looks more like a staircase or zig-zag pattern rather than a flatline (see Figure 7.2). On a side note, all signals are *standardized*, which means that the amplitudes are relative and, therefore, varies between the signal captures.

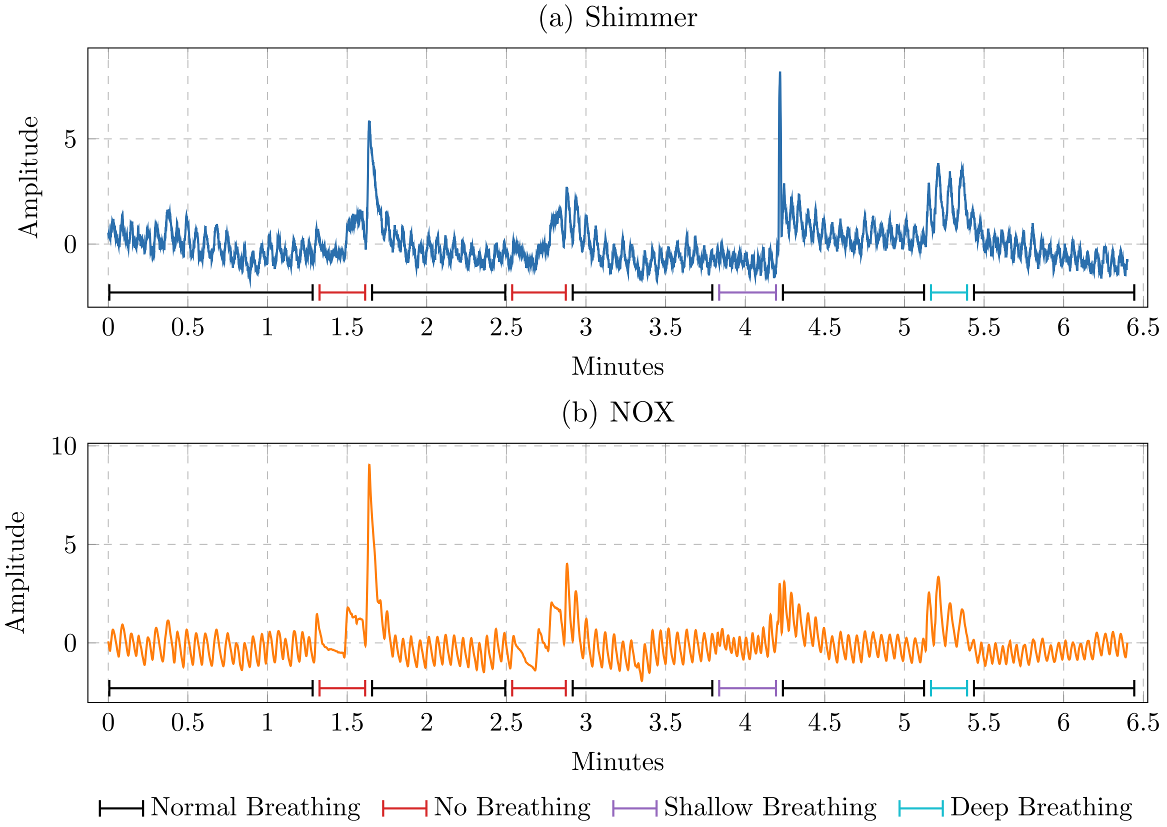
Three examples of Shimmer captures from three different subjects are shown in Figure 7.1, Figure 7.2, and Figure 7.3. The signal quality in these examples is (by the metrics) rated as *good*, *typical*, and *bad*, respectively. For the good quality capture (Figure 7.1), the waveforms of Shimmer and NOX are almost identical. Visually, the only thing that separates them is a slight baseline wander. The high-frequency current inflicted to the subject's skin by Shimmer (Section 3.5.1) is visible during the complete breathing stops for all signal captures. Whereas the NOX produces a very clean flatline, the unfiltered signal from Shimmer can never be as flat because of the inflicted current. As seen in the more typical signal quality capture (Figure 7.2), the ratio between breaths and this inflicted high-frequency current is lower compared to the good quality capture (Figure 7.1). In Figure 7.2, the waveforms are still very similar to the NOX and the periods of disrupted breathing are also very distinguishable from normal breathing, but it is also slightly harder to distinguish noise from breaths in general. In some cases, the signal quality from Shimmer is somewhat poor. In Figure 7.3, it is almost impossible to differentiate between the periods of no breathing and shallow breathing, and many normal breaths are totally overwhelmed by noise.

Likewise for BITalino, three examples of captures from three different subjects are shown in Figure 7.4, Figure 7.5, and Figure 7.6 rated as *good*, *typical*, and *bad*, respectively. The first matter one may notice is that even for the good quality BITalino capture, the waveforms are not nearly as identical to the NOX as the good and typical Shimmer captures are. The reason for this is that both Shimmer and NOX captures the respiratory process as *volume*, whereas BITalino as *airflow*. In other words, all three examples presented here are *raw* signals, which are only standardized, synchronized, and downsampled. Even despite the visual looks of it, the good quality BITalino capture shown in this example outperforms the good quality Shimmer capture shown in Figure 7.1 for both breath detection as well as amplitude accuracy. As the signal quality of BITalino degrades, the noise gets more prominent, making breaths harder to distinguish (see Figure 7.5 and Figure 7.6). A rather major noise factor for these signals (also Shimmer) is the heartbeats (see Figure 7.7), which for many of the subjects show an amplitude more than 10% of the mean breath amplitude. The frequencies and behavior of breaths and heartbeats are quite different and are, therefore, in many situations trivial to distinguish. Nonetheless,


(a) Shimmer

(b) NOX

Figure 7.1: Example of a very good quality Shimmer capture


(a) Shimmer

(b) NOX

Figure 7.2: Example of a typical quality Shimmer capture

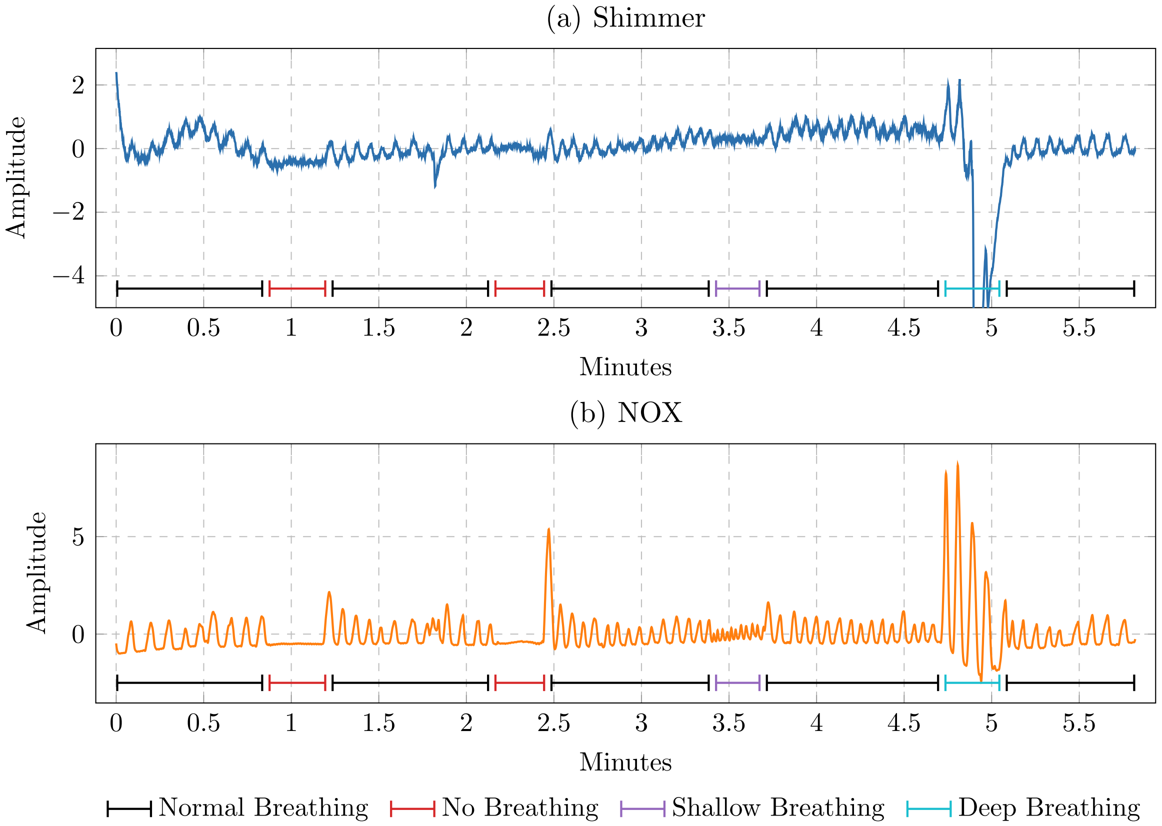


Figure 7.3: Example of a somewhat bad quality Shimmer capture

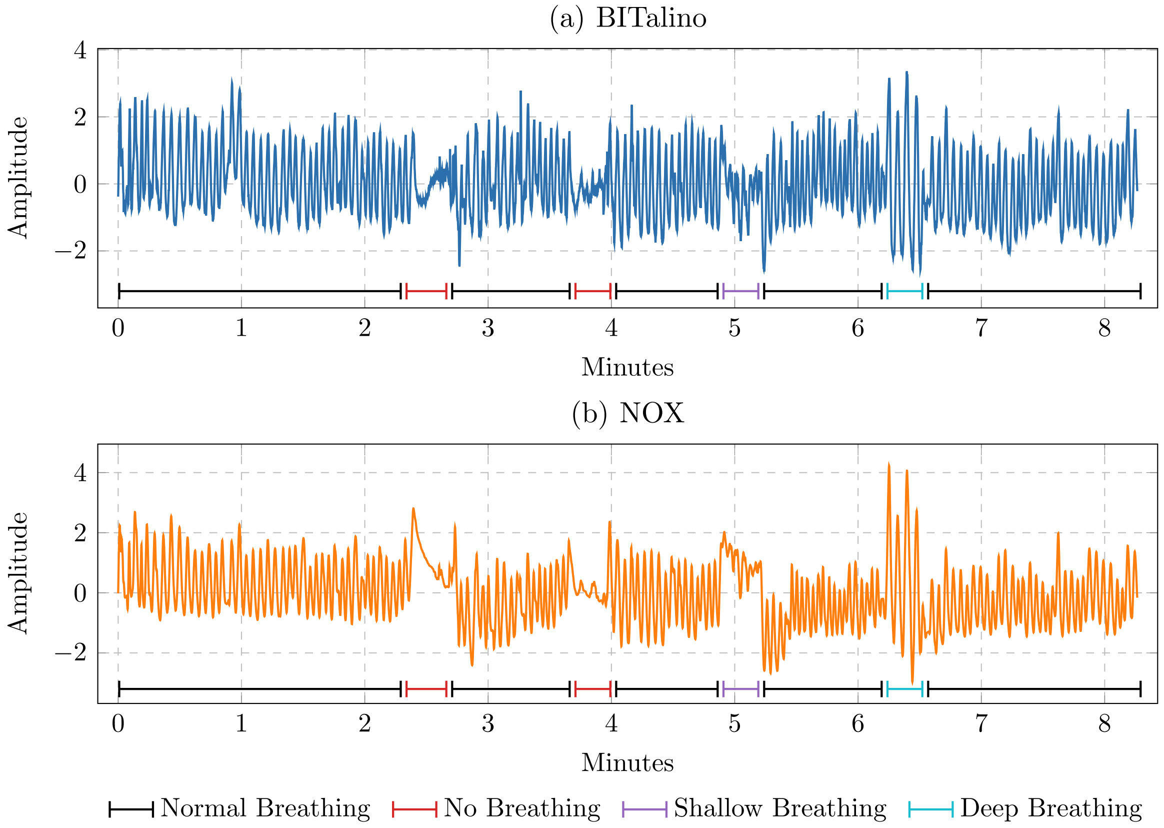
as the signal-to-noise ratio declines, heartbeats can often easily be mistaken for shallow breaths.

For reference, one may notice how clean the gold standard signal (NOX) is compared to the signals from both BITalino and Shimmer. The NOX signal does contain a slight baseline wander, but high-frequency noise is almost non-existent. The amplitude of heartbeats is extremely low, making the breaths the (almost) sole component of the signal. Even the most shallow breaths are easily identifiable.

7.3 BITalino

7.3.1 Noise Removal Procedure

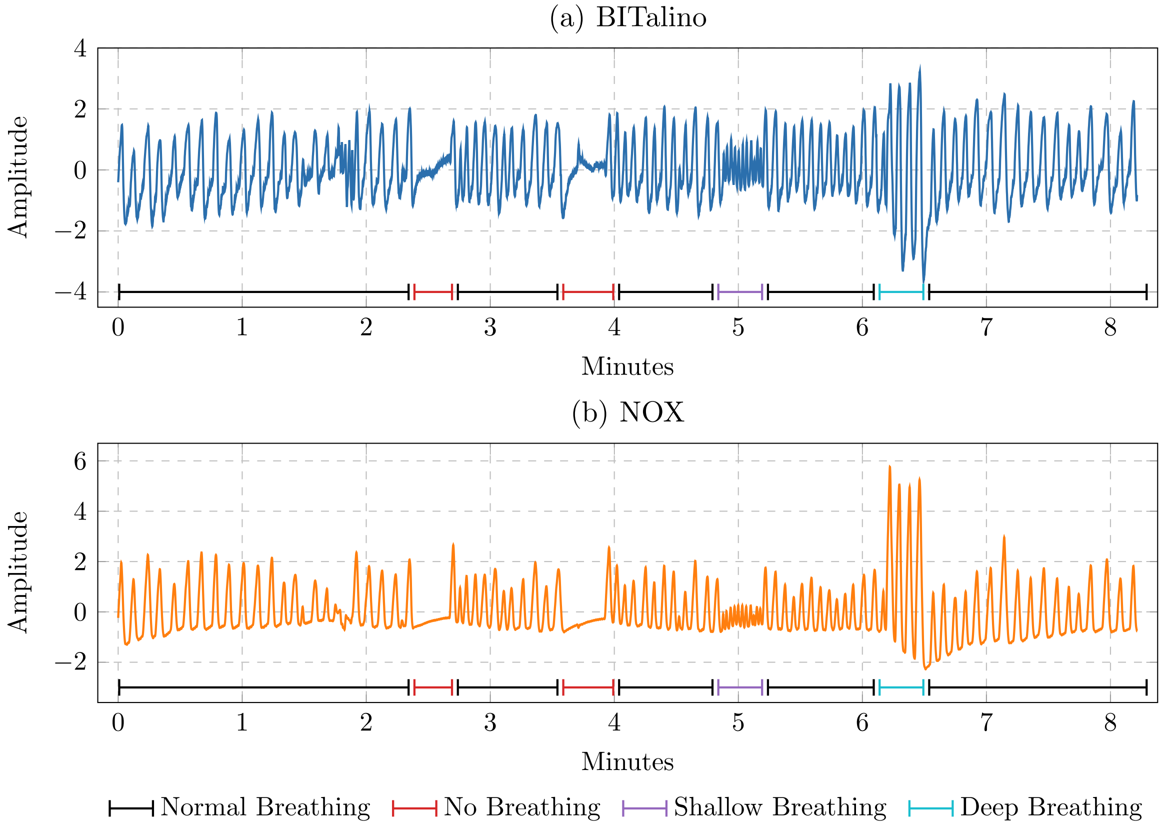
During the preliminary testing described in Chapter 5, we discovered a total of four issues or oddities with the BITalino PZT effort belts: (1) The signal randomly flipped across its y-axis (i.e., turned upside down), (2) Body position changes resulted in drastic changes to the baseline breath amplitude, (3) The signal randomly got stuck near its minimum or maximum values, and (4) The signal was sometimes heavily contaminated by white noise. Before we gathered any signal captures from the subjects, we were able to find a workaround for the latter two issues (3 and 4). The issues seem to be caused by static electricity of some kind, and BITalino themselves confirmed that the problem is caused by a production fault affecting this specific batch of devices. Nevertheless, we performed



(a) BITalino

(b) NOX

Figure 7.4: Example of a very good quality BITalino capture



(a) BITalino

(b) NOX

Figure 7.5: Example of a typical quality BITalino capture

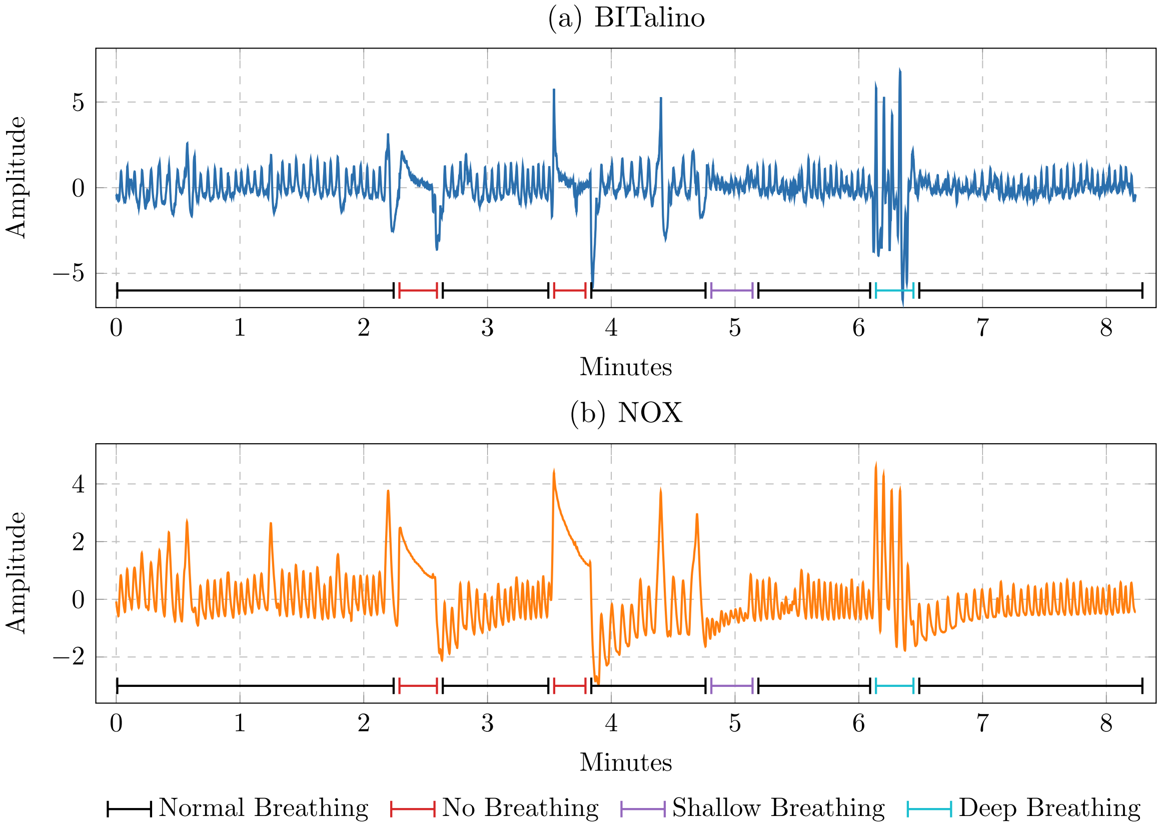


Figure 7.6: Example of a somewhat bad quality BITalino capture

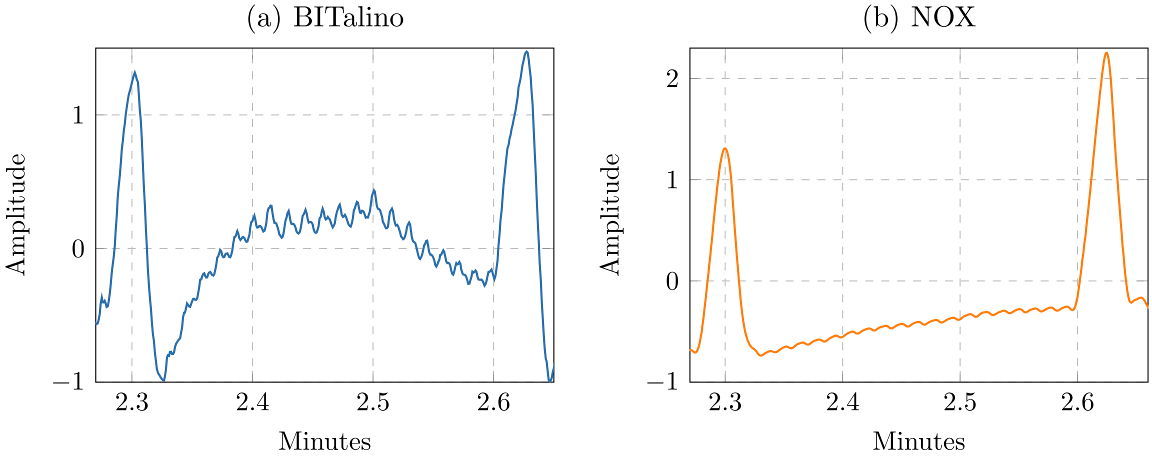


Figure 7.7: Heartbeats visible during a complete breathing stop

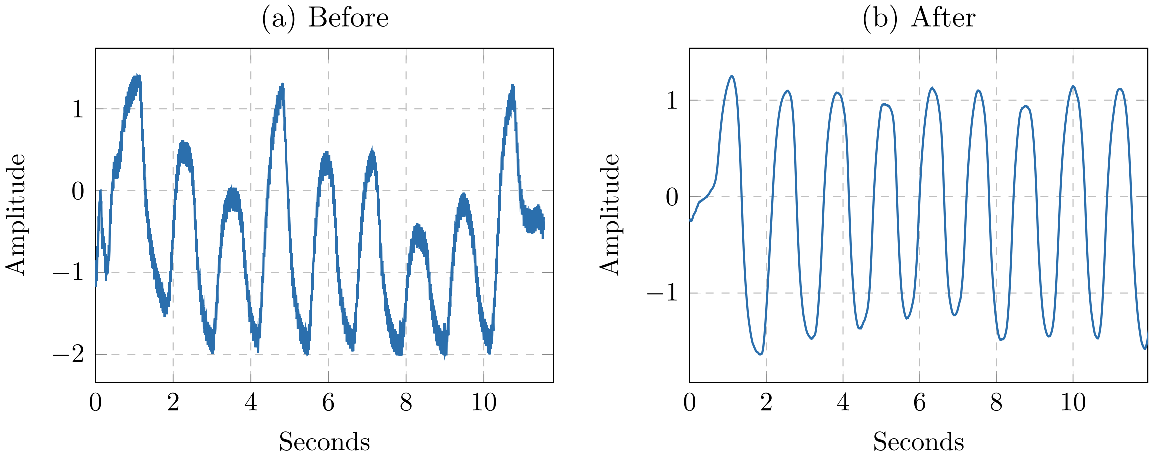


Figure 7.8: Comparison of BITalino before and after the noise removal procedure

the following workaround (and confirmed its effectiveness) before we captured data from all subjects:

1. Unplug the BITalino battery.
2. Plug and unplug the USB cable between the BITalino and a PC (with a connected charger) a few times.
3. Unplug the USB cable and replug the BITalino battery.

The effectiveness of the procedure can be seen in Figure 7.8. This workaround did not only remove the white noise, but the breath amplitudes also became more deterministic in relation to belt distraction. Moreover and most importantly, the signal no longer got stuck near its minimum or maximum values no matter how loose or tight the belts were fitted around the subject.

7.3.2 Integrating Noisy Signals

We described in Chapter 5 that the BITalino PZT belts capture *airflow* rather than *volume* and, therefore, needs to be integrated before it can be compared to the NOX. However, it turns out that the signal is too affected by noise and measurement errors, resulting in a greater distance to the gold standard after integration, compared to the raw signal. As shown in Figure 7.9, the ratio between breath amplitudes in an airflow signal remains after integration *only* for those breaths with an equal duration. In other words, comparing the raw BITalino signal against the NOX would only be accurate if all breaths were of equal duration. Yet, it turns out to be more accurate than integrating the signal.

The increased distance of the integrated signal is mostly related to deeper breaths, whose amplitudes are consistently too low compared to the NOX. Any measurement errors and noise regarding the duration of breaths affect the accuracy of the amplitudes of the integrated signal negatively. As specified in the design, we filter the raw signal with a Butterworth high-pass filter with a low-cut frequency of 0.1 Hz before integration.

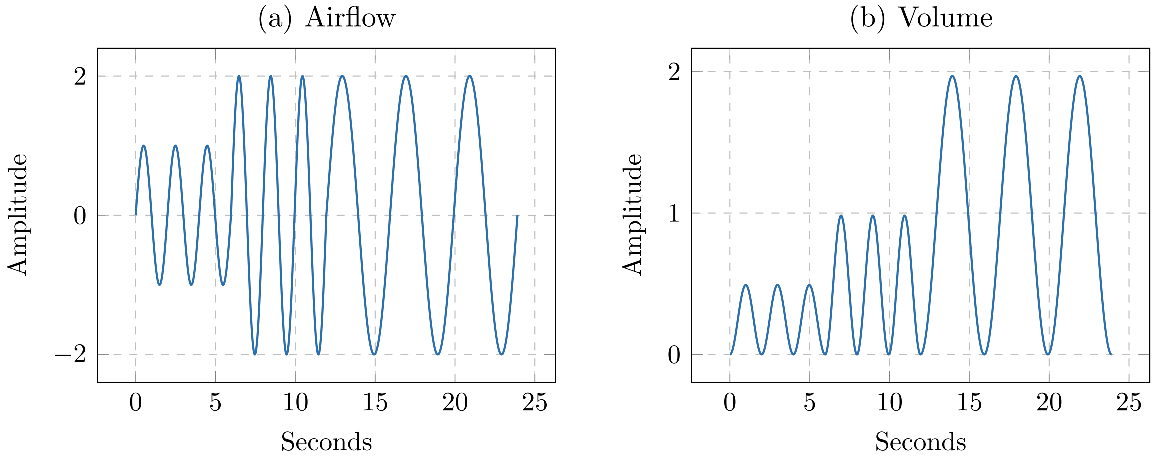


Figure 7.9: Synthetic example of airflow and volume amplitude relationship

Adjusting this cut-off frequency, or applying a bandpass filter instead, actually worsens the result further. In fact, integration by itself is after all a low-pass filter. Interestingly, excluding the few largest amplitude breaths from the metric calculation does *sometimes* make the integrated result *slightly* better than the result for the raw signal. Consequently, we also measure the signal quality of the raw signals in addition to the integrated signals alone.

7.3.3 Signals

From each subject, there are a total of four different raw signals from BITalino, the abdominal and thoracic belt signals from both the supine and side body positions. From each body position, we also generate the $\mathrm{PZT_{sum}}$ signal ($\mathrm{RIP_{sum}}$ for NOX) by summing the abdominal and thoracic signals. Furthermore, we also integrate all these signals, resulting in a total of *twelve* different signals from BITalino per subject.

For each signal quality metric, we present the three ”raw” signals (abdominal, thoracic, and sum) from the supine position separately from the same signals from the side position; and their integrated counterparts separately in the same manner. As a result, we present four charts per metric, each containing the results of three signals. We present only visual charts in this chapter, but the raw data can be found in Appendix B.

Both BITalino signals from Subject 10 for the side position are corrupt and contain no meaningful data, and are, therefore, excluded from the results. The subject did not lie directly on the sensor part of the belts, neither were the belts fitted too tight or too loose, and so the cause remains unknown. The same applies to the thoracic signal from the side position of Subject 4, which is also corrupt. An example of a corrupt signal from BITalino is shown in Figure 7.10. The only feature that is identifiable in this example is the deep breaths. We declare a signal corrupt if the majority of the signal is somewhat like the example shown in Figure 7.10.

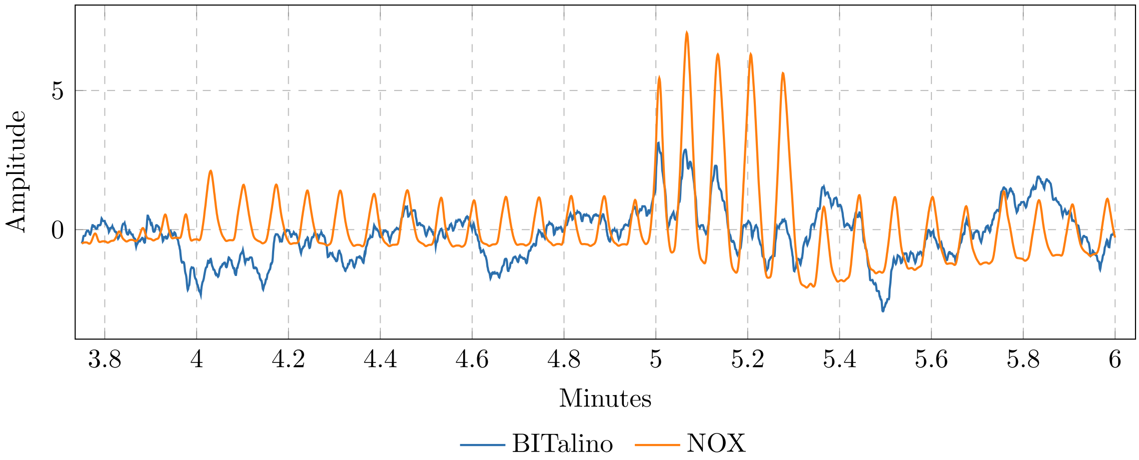


Figure 7.10: Example of a corrupt BITalino capture

7.3.4 Breath Detection Accuracy

Sensitivity

Sensitivity describes the proportion of correctly identified real breaths, and a sensitivity of 100%, thus, implies that all real breaths are correctly identified. The total number of real breaths for each of these signal captures lies in the range of 80–130, which means that each missing breath results in a sensitivity loss of about 1.25–0.76%. One must note that sensitivity alone does not directly imply that the signal is of good quality. A large number of false breaths increases the odds of false breaths being identified as true, and may, thus, also result in a higher sensitivity. Whereas a large number of false breaths may increase sensitivity, it may also make it significantly more challenging to detect epochs of disrupted breathing.

As shown in Figure 7.11a and Figure 7.11b, the sensitivity of the raw BITalino signals are very good for both positions, with a value above 95% for eleven out of twelve subjects in the supine position, and ten out of eleven subjects in the side position. The exception is the thoracic signal, which is slightly better and more stable overall in the side position compared to the supine position (excluding the corrupt signals). The breath amplitudes of the thoracic signal are for the majority of the subjects much lower compared to their abdominal counterparts. This results in a lower signal-to-noise ratio, making it harder to distinguish breaths from noise in general, and explains the lower sensitivity of the thoracic signal. The sum of the abdominal and thoracic signals amplifies features that are common to both signals (i.e., breaths) and minimizes the features that are unique to one of the signals (i.e., noise). The expected outcome is that the sum-signal should perform better than the abdominal and thoracic signals alone. However, whereas the sum-signal is overall very good, it is also rarely better than both the *raw* abdominal and thoracic signals but lies more often somewhere between the two.

Integration acts as a low-pass filter and, therefore, smoothes the signal. Hence, low amplitude noise is minimized, but so are low amplitude breaths. As shown in Figure 7.11c and Figure 7.11d, the sensitivity of most of the integrated signals are lower compared to their raw counterparts. This result implies that the signal-to-noise ratio of the affected

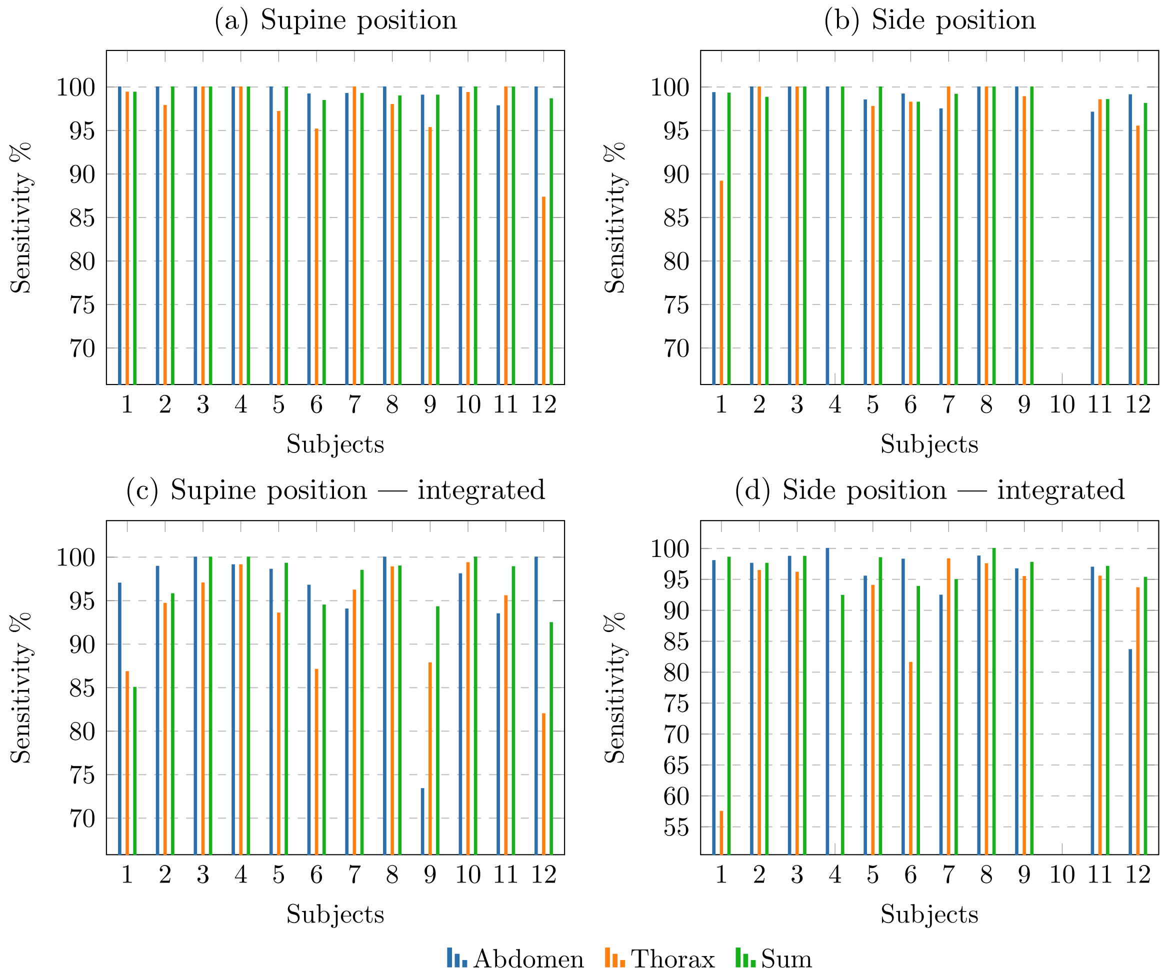


Figure 7.11: Sensitivity of all signals from BITalino

signals are low to begin with, and the low amplitude breaths are attenuated during the integration, resulting in a lower sensitivity. For most of the subjects, there is higher sensitivity loss for the thoracic signal compared to the abdominal, which further substantiates this observation. One may notice that the sensitivity of the sum-signal is less affected by the integration compared to the abdominal and thoracic signals. In this case, the sum-signal is better than both the abdominal and thoracic signals for six out of twelve and five out of eleven subjects in the supine and side positions, respectively.

Positive Predictive Value

The positive predictive value (PPV) describes the proportion of detected breaths which are *real breaths*. A PPV of 100% means that all detected breaths are real breaths, and analogously, a PPV of 90% means that 10% of the detected breaths are false breaths. Unlike sensitivity, the PPV does not decrease linearly as false breaths are added to a signal because the number of false breaths is a part of the denominator rather than the numerator. Given, for example, a signal with 100 real breaths. Adding one false breath to this signal causes a PPV decrease of 0.99%, while adding 100 false breaths causes a decrease of 50%.

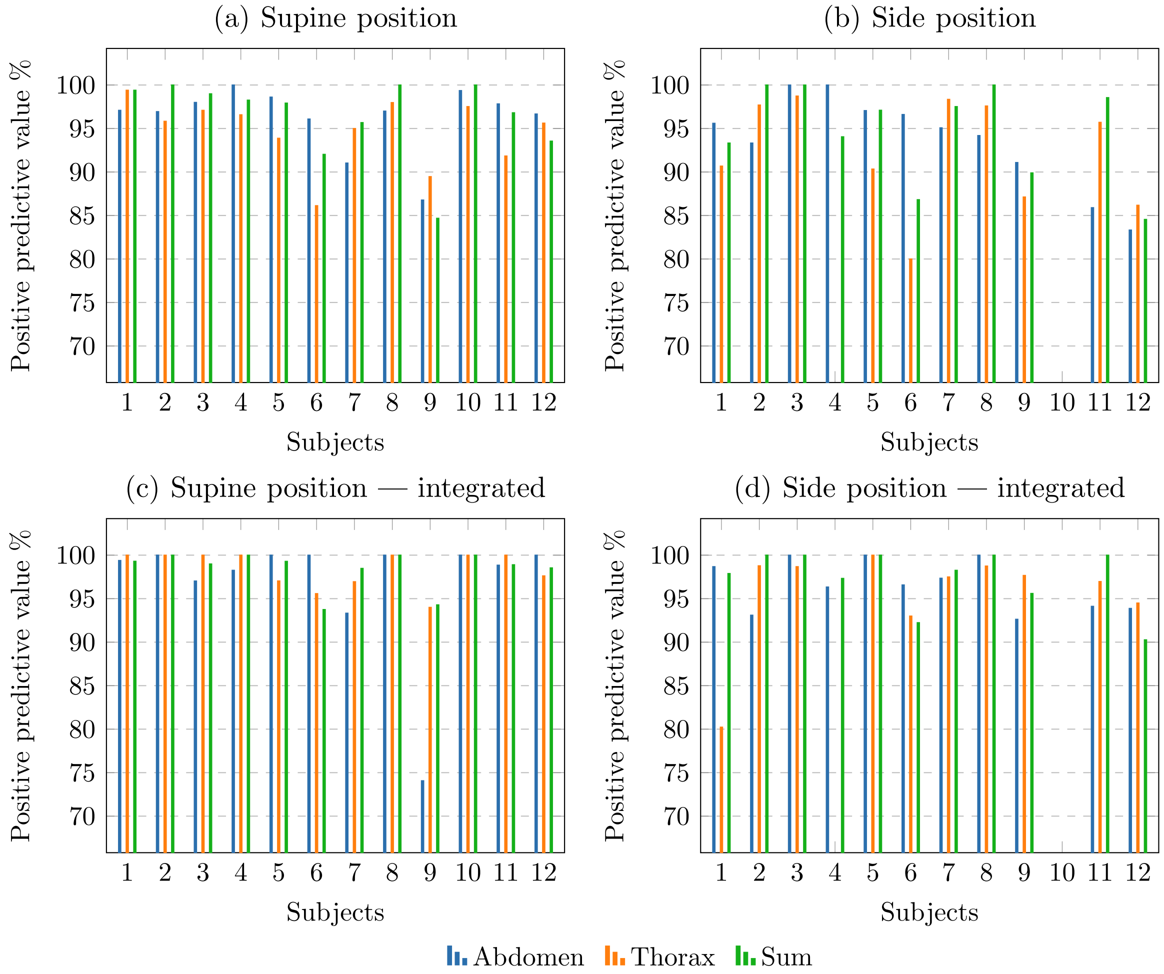


Figure 7.12: Positive predictive value of all signals from BITalino

Interestingly, integration has the opposite effect on PPV as opposed to sensitivity (see Figure 7.12). After integration, the PPV is improved for almost all signals from all subjects, for both the supine and side positions. In fact, of 68 signals, only ten signals show a decreased PPV after integration. This result implies that the false breaths present in the raw signals are of low amplitudes, which are then attenuated during the integration. The PPV of the sum-signal improved on average the least by the integration. Whereas random noise is attenuated during the summation, the amplitude of heartbeats and other common features remain mostly unaffected as they are present in both the abdominal and thoracic signals. The amplitudes are larger in the sum-signal, which explains why most features sustain the integration process.

In general, the PPV for most signals is worse and less stable than the sensitivity. For the raw signals, ten out of twelve subjects show a PPV above 90% for all signals in the supine position, while seven out of eleven subjects show the same in the side position. In conclusion, up to 20% of all detected breaths in the raw signals, and up to 10% in the integrated signals (excluding the outliers) are, in fact, false breaths.

Clean Minute Proportion

The clean minute proportion (CMP) describes the proportion of minutes in the signal where both the sensitivity and PPV are 100%. These signals are about seven minutes in duration, which means that one dirty minute results in a CMP decrease of about 14%. The CMP by itself may not be a very interesting metric, but combined with sensitivity and PPV, it explains the distribution of errors. If, for example, a signal has a low PPV, the CMP yields information about whether the false breaths are spread throughout the signal or contained within a few minutes.

There is a very noticeable correlation between the PPV and CMP values of the signals. In fact, for most of the signals, a low PPV also directly corresponds to a low CMP. See, for example, Subject 5–9 between Figure 7.12a and Figure 7.13a. This suggests that most of the false breaths are, in fact, distributed somewhat evenly throughout the signals. Given that the sensitivity of all the signals is very good, the correlation between sensitivity and CMP is less noticeable, although still present.

While integration attenuates low amplitude features from the signal, regardless of those being true or false breaths, the CMP value is often better after the integration. Given that false breaths are the biggest factor affecting the signal quality of these sensors, this is as expected. Those signals whose sensitivity decreased the most after integration, are also the ones whose CMP decreased after integration. In other words, whether the CMP of a signal decreases or increases after integration, depends upon which of PPV and sensitivity that are most affected by the integration. This observation suggests that not only are false breaths evenly distributed but so are the missing real breaths.

7.3.5 Breath Amplitude Accuracy

The breath amplitude accuracy for all subjects is presented in Figure 7.14. The WAPE metric calculates the *error* (or *distance*), which means that the lower the metric score, the higher the accuracy. In other words, *lower is better.* One must note that the result of entirely random data for this metric is 50%, which means that anything close to or worse than this may correspond to an inferior performance depending on the underlying distribution.

The first matter one may notice is that the thoracic signal is significantly worse than the abdominal. It is, in fact, more than twice as bad for Subject 2–6 and Subject 12 in the supine position. Overall, it is worse than the abdominal signal for nine out of twelve and seven out of ten subjects for the supine and side positions, respectively. In the few other cases, it is either equal to or only slightly better than the abdominal signal. The exception is Subject 1, for which the abdominal belt did not fit properly, resulting in reduced performance in the supine position. For the side position, however, the non-optimal fit of the belt does not affect the performance very much as it achieves an average score.

For the raw signals, the accuracy of the abdominal signal is slightly better and more stable for the supine position compared to the side position. Including all subjects, the mean amplitude error is 13.8% and 16.5% for the supine and side positions, respectively.

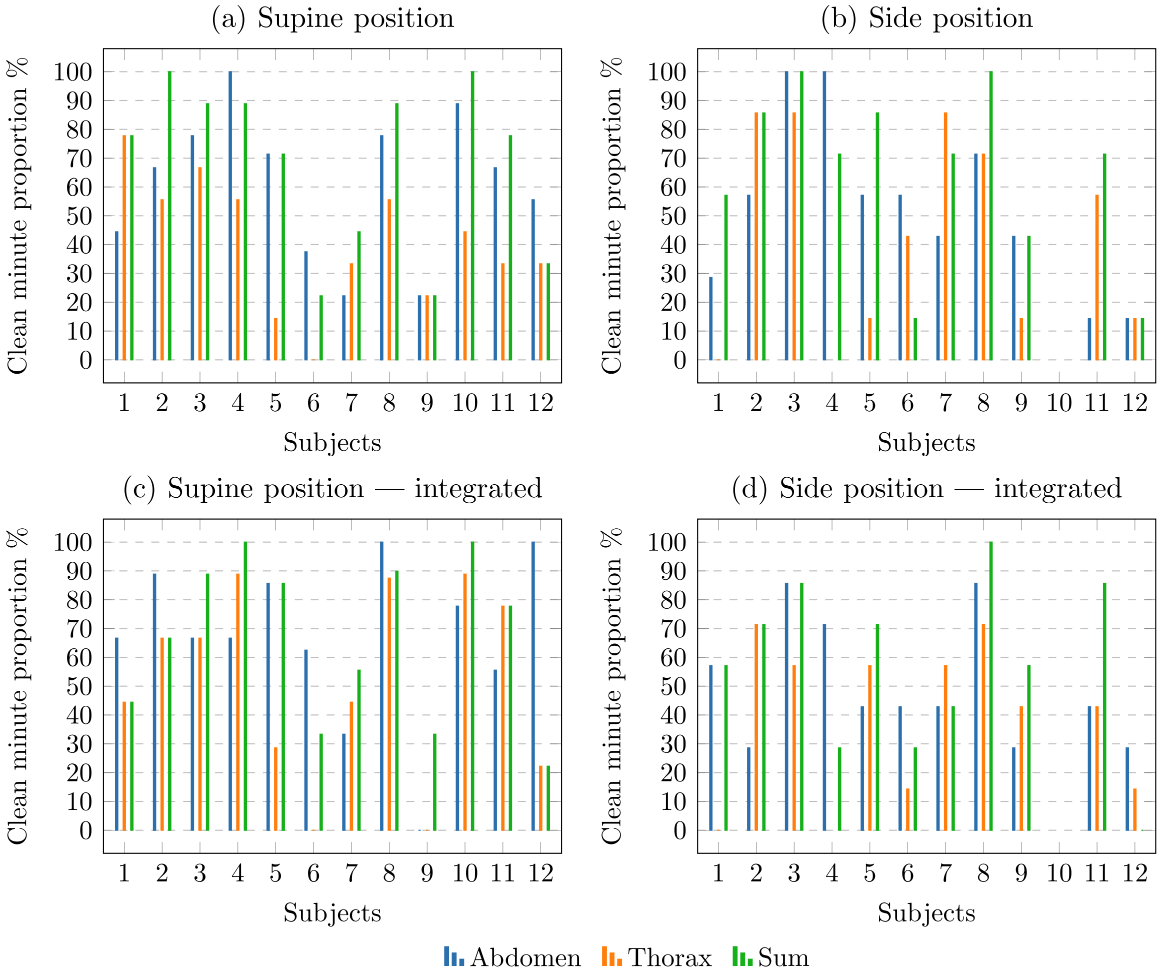


Figure 7.13: Clean minute proportion of all signals from BITalino

The central tendency of the abdominal signal (excluding outliers) is located in the range of 5–15% for the supine position, as eight out of twelve subjects lie within this range (Subject 2–6, 8, 10 and 12). For the side position, the central tendency lies in the range of 10–20% (Subject 1, 3–6 and 9).

The results of integration show a decrease in accuracy for all signals. The exception is only one signal (Subject 5 raw abdomen side) which shows a *very* slight increase. After integration for both positions, the best signals show a decreased accuracy of about 5%, whereas the signals with worse accuracy show an even greater decrease in accuracy. We suspect that the cause behind this phenomenon is mainly related to measurement errors regarding both the amplitude and duration of breaths. These kinds of measurement errors amplify during the integration, resulting in a decrease rather than an increase in accuracy. As shown in Figure 7.15, the measurement error regarding breath duration is very visible for the highest amplitude breaths. In this example, the signal exhibits a sudden decrease in airflow in the midst of all the deep breaths, which is clearly wrong, splitting them into two breaths of lower amplitude. In the integrated version of the signal, the normal amplitude breaths match the NOX better than their raw counterparts, but the deep breaths, however, show a much greater distance. This behavior is very common amongst all signals from all subjects, but a few signals are unaffected.

The raw abdominal signal from Subject 12 shows the best accuracy of all signals with a score of 5.65%. Interestingly, out of nine signals with a score below 10%, only three sustain a score below 10% after integration (Subject 4 and 12). Figure 7.16 shows a scatterplot of the breath amplitude relationship between BITalino and NOX, including the regression line, for a select few of the raw abdominal signals from the supine position. The selection is based on all the different kinds of relationships present in the data and shows how such relationships affect quality. The sub-figures are sorted based on quality, with Figure 7.16a, Figure 7.16b, Figure 7.16c, and Figure 7.16d having a score of 5.65%, 9.68%, 16.93%, and 20.95%, respectively. Figure 7.16a shows a linear relationship between the normal and deep breaths, while the amplitudes of the shallow breaths are slightly too low, making the overall relationship more monotonic. The fact that both the normal and deep breaths are aligned this close to the regression line is what causes the signal to score this well. Figure 7.16b and Figure 7.16d show an evident monotonic relationship when the deep breaths are included but in the opposite direction of each other. Figure 7.16c exhibits more variation compared to the others, and while it may be harder to see, this signal does also indicate a monotonic relationship. If the shallow breaths were removed, the slope of the regression line would be steeper. As seen from these examples, the deep breaths have the largest room for errors, which results in a significant influence on the metric score.

7.3.6 Conclusions

The primary concern regarding these sensors' breath detection accuracy is the low signal-to-noise ratio. While the sensitivity of all signals is very good, the PPV is generally not. All signals, including the integrated versions, are affected by a somewhat large number of false breaths. Whenever the signal-to-noise ratio is better, integration is an effective method to minimize the number of false breaths. However, as a good signal-to-noise ratio is generally not the norm, integration does more harm (to sensitivity) than good

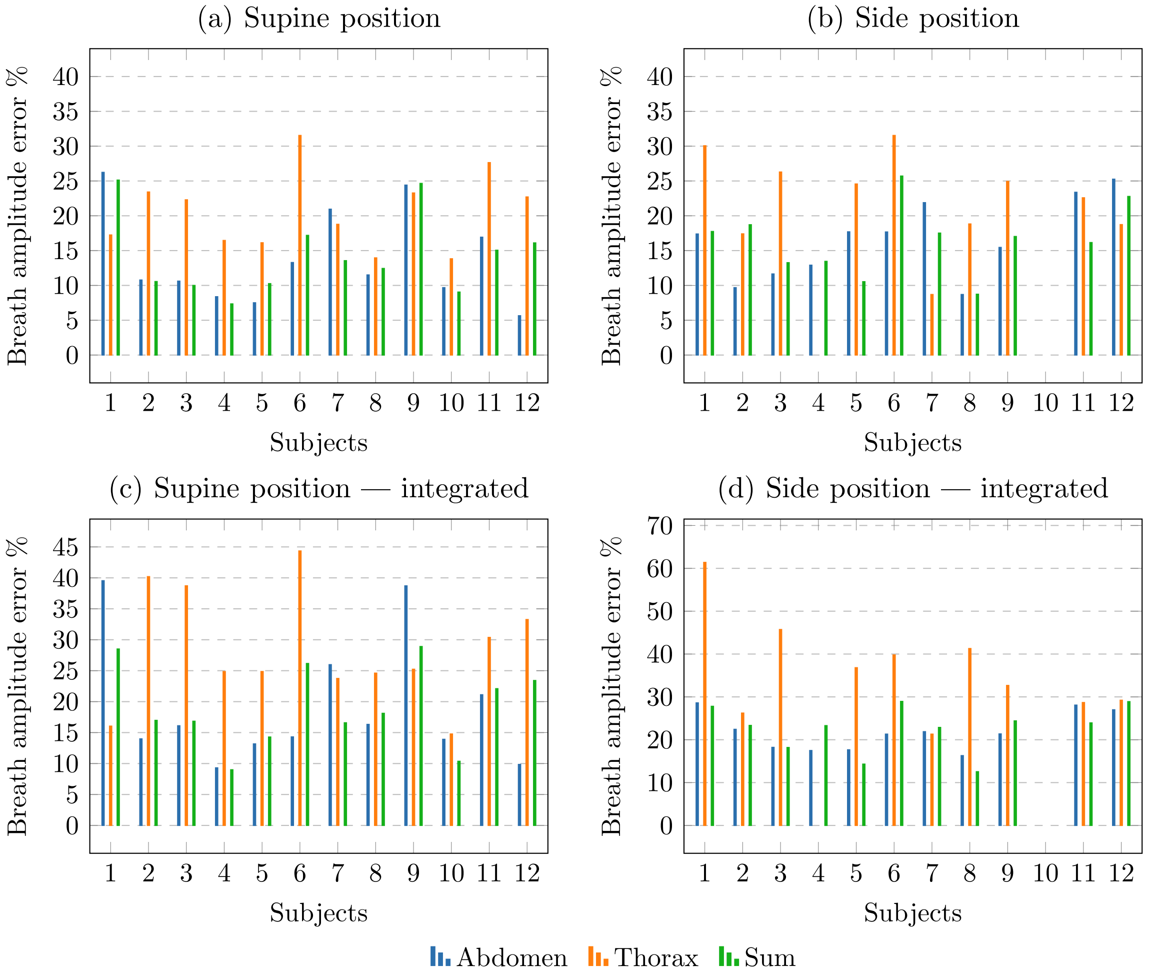


Figure 7.14: Breath amplitude error of all signals from BITalino

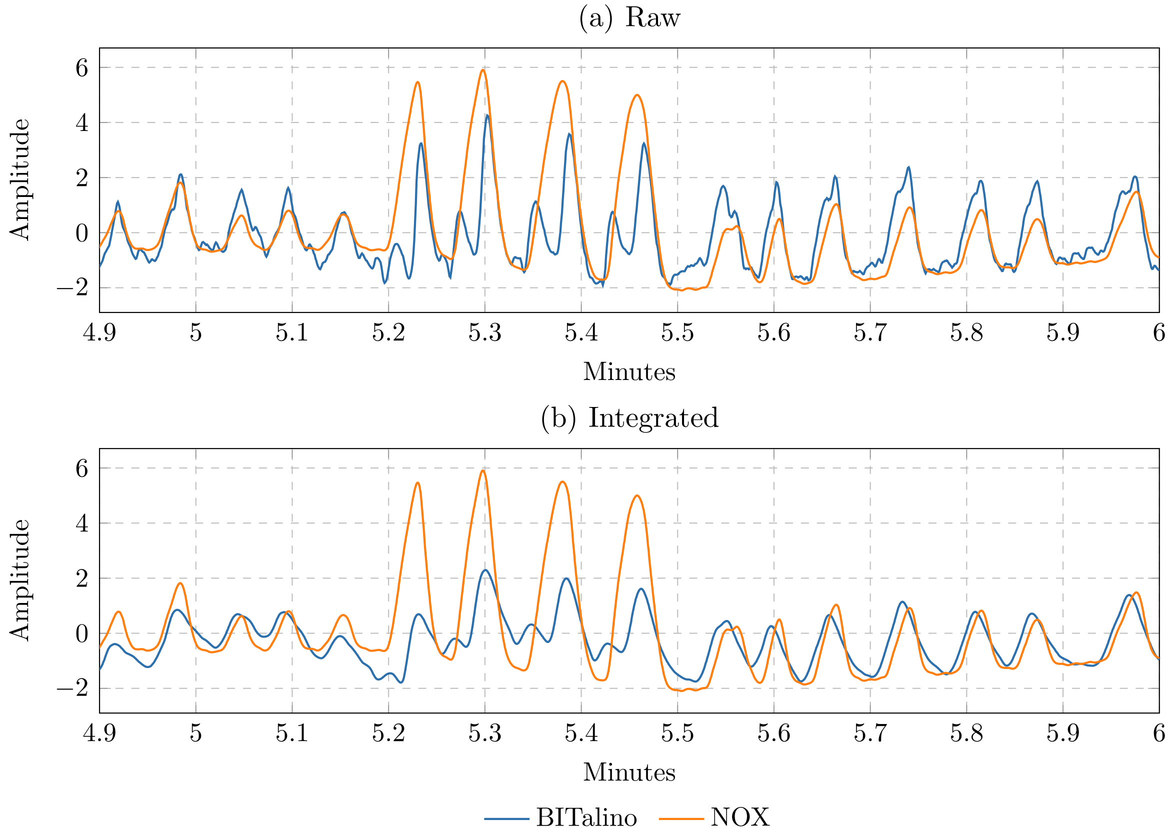


Figure 7.15: Measurement error of breath duration — Subject 2 abdomen side

(to PPV).

Regarding apnea detection, the raw signals perform very poorly if the scoring rules by the AASM are followed strictly. These scoring rules state that there has to be no signal excursion of more than 10% of the baseline breath amplitude for at least ten seconds, for a period to be regarded as apneic. The raw signals for these sensors simply never produce a line which is nearly flat, as the heartbeats and other kinds of noise are constantly showing an amplitude greater than 10% of the baseline. To follow the AASM rules strictly, one would either have to recognize such features for what they are or carefully filter them out of the signal. Heartbeats are, for example, very effectively attenuated by integration due to their short duration compared to breaths.

The relationship of the breath amplitudes between BITalino and NOX are often monotonic, rather than linear, and the slope of the relationship is not consistent across different subjects or signals. By regarding the relationship of the normal breaths as the frame of reference, it is either the shallow breaths, deep breaths, or a combination, which reduces the linear relationship into a monotonic one. With no doubt, the deep breaths deviate the most from the linear relationship. Since the detection of hypopneas is mostly concerned with the transition from normal to shallow breathing, these sensors may therefore still be very much adequate for the purpose. When including all three types of breaths, the abdominal signal shows on average a breath amplitude error of 13.8% in the supine position. This means that a 30% reduction in airflow is on average recorded by the sensor

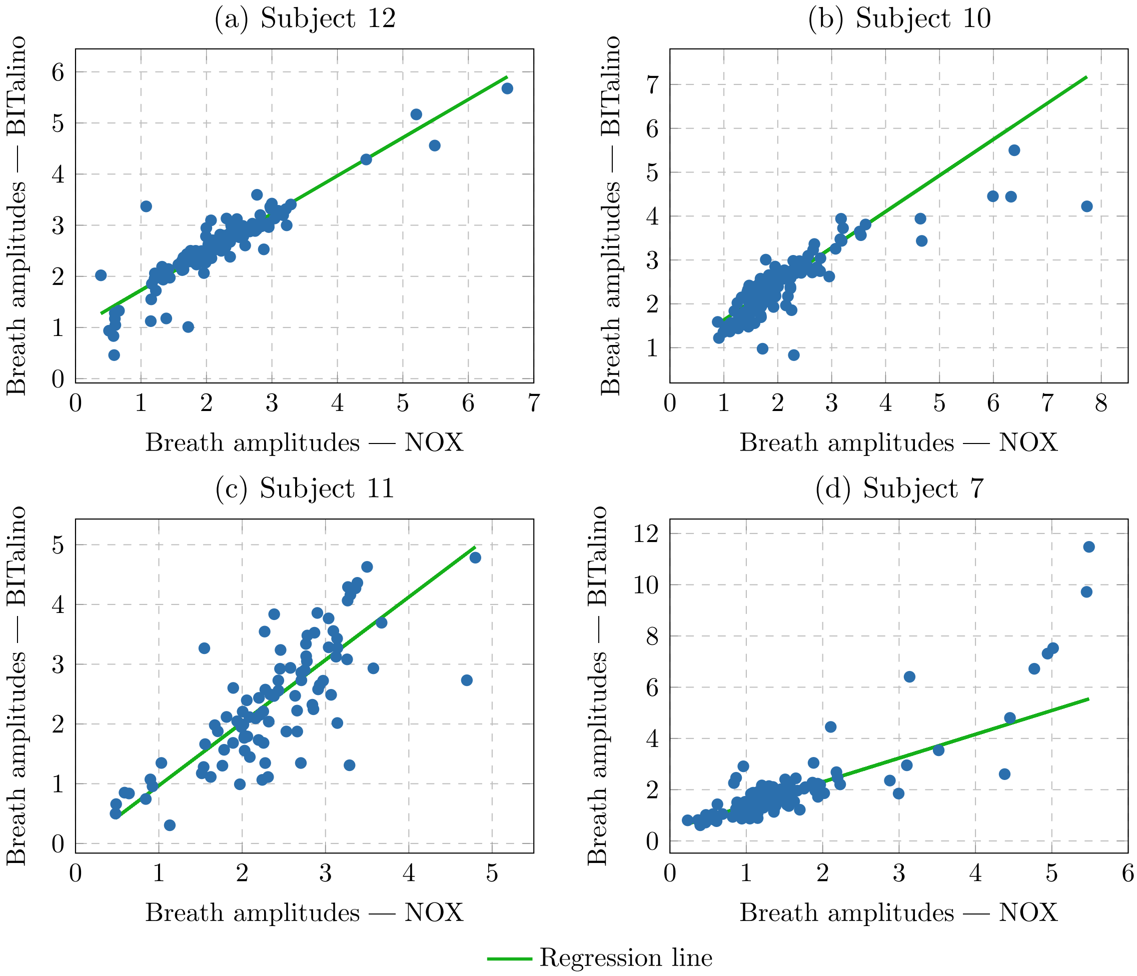


Figure 7.16: Breath amplitude relationship between BITalino and NOX (supine, abdomen, raw)

as a value in the range of 16.2–43.8%. As a result, hypopneas with a reduction in airflow greater than 43.8% are very likely to also be correctly identified as such.

There are no clearly visible trends in the data related to BMI and signal performance. In the supine position, both subjects with the highest and lowest BMI show the best amplitude accuracies overall for the abdominal signal. Although, there may be an ever so slight trend in the abdominal signal for the side position. The breath amplitude accuracy is steadily decreasing in the interval from Subject 2–7 and Subject 11–12. The same signals' PPV is also steadily decreasing in the interval from Subject 4–12 (although Subject 10 is missing).

We discovered an issue during the preliminary testing where the BITalino signal would suddenly flip across its y-axis. This phenomenon turns out to be very common. In fact, at least one of the raw signals (abdominal or thoracic) from most subjects is affected. Identifying such periods is not always trivial. Nonetheless, we have come to the conclusion that it may not be too much of concern regarding apnea detection. We do not expect periods of disrupted breathing to be more challenging to identify in a flipped signal; at least not for humans.

In conclusion, the signal quality of the abdominal signal is superior, with the sum-signal not far behind. During breathing obstructions, the abdominal signal may still show significant signal excursion due to respiratory effort (paradoxical breathing). Therefore, it is unclear whether the abdominal signal alone is sufficient to reliably detect apneic and hypopneic episodes, or if the thoracic signal is needed to generate the sum-signal. On the other hand, Kristiansen et al. (2018) show excellent classifier performance using the abdominal signal alone when the data is of good quality. They do, however, not evaluate the sum-signal, and it is, therefore, still unclear whether the abdominal signal alone is sufficient, especially for lower quality signals.

7.4 Shimmer

7.4.1 Signals

In contrast to BITalino's twelve signals per subject, Shimmer provides only two signals per subject. Shimmer records just the thoracic expansion and contraction, not abdominal, which results in one signal from each of the two body positions. As of this fact, we present all the four metrics for Shimmer together in Figure 7.17, where each of the four metrics is separated into their own sub-figures containing the results of both body positions.

A significant number of the signals from Shimmer are, in fact, corrupt. Of all the subjects with a BMI above 30, three out of four signals from the side position are corrupt, whereas none of the side position signals from the subjects below 30 BMI are corrupt. The affected signals are overwhelmed by noise, and it is impossible to even distinguish a single normal breath (deep breaths are still *sometimes* distinguishable). The same phenomenon occurred only for one subject (Subject 8) in the supine position. We double checked the electrodes after the sessions, and they were still properly attached, which means that the cause is at least not related to electrode slippage. All the affected signals are excluded

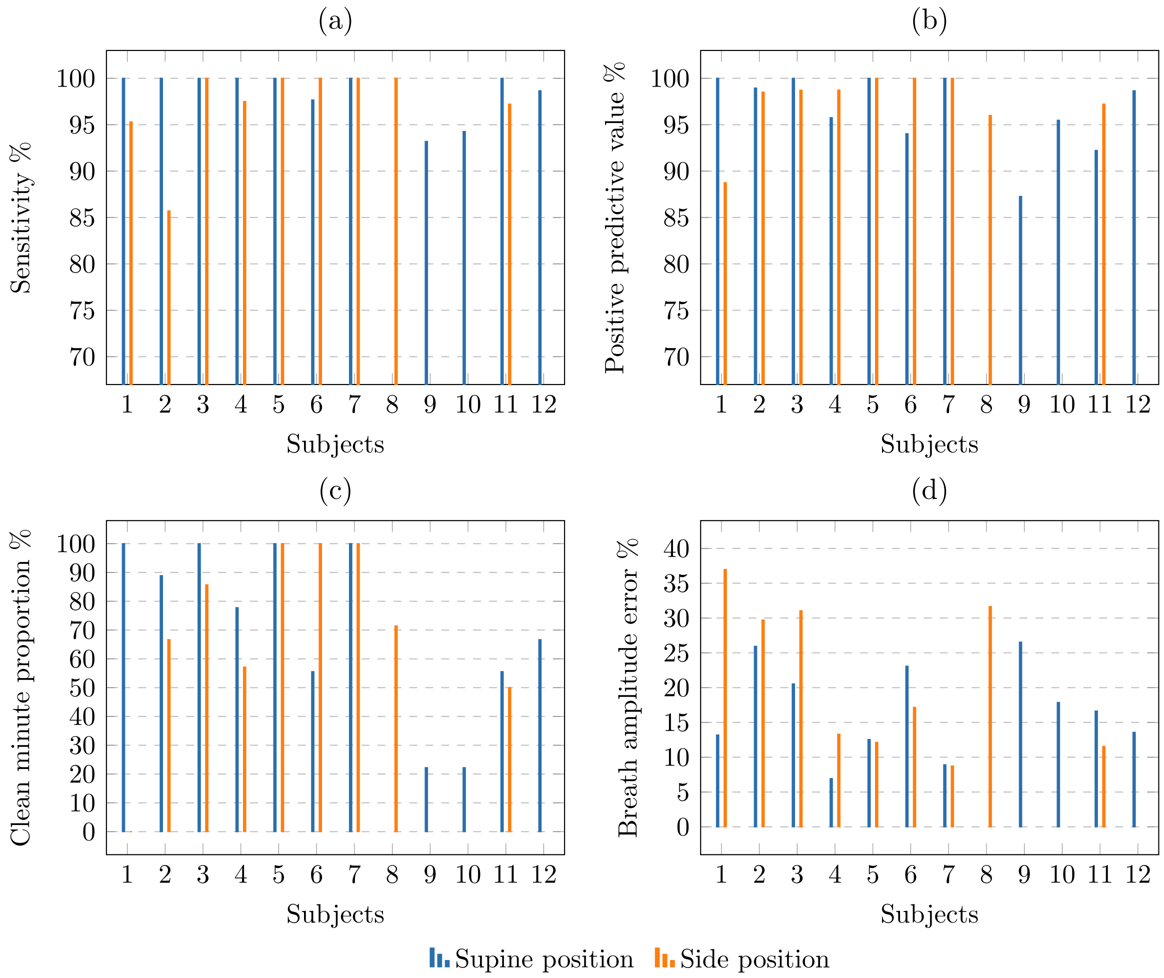


Figure 7.17: All quality metrics for all signals from Shimmer

from the results, but nonetheless, the observation in itself remains significant. An example of a corrupt signal from Shimmer is shown in Figure 7.18. As seen, without using the NOX signal as a reference, it is impossible to distinguish most breaths. We declare a signal corrupt if the majority of the signal is as the example shown in Figure 7.18.

7.4.2 Breath Detection Accuracy

Figure 7.17a presents the sensitivity of both body positions from Shimmer. The first point one may notice is that when the sensitivity is good, it is very good. In contrast to the sensitivity of BITalino where the results are often close to 100% but not exactly 100%, the sensitivity of Shimmer is often either precisely 100%, or significantly worse (with a few exceptions). Even without considering the corrupt signals, the sensitivity from the supine position is a little better and more stable overall compared to the side position. The mean sensitivity of the supine and side positions is 98.53% and 97.30%, respectively.

For the PPV (Figure 7.17b), the signals with a perfect sensitivity mostly also have a perfect or almost perfect PPV. The exception is the signals from Subject 2. The side position signal from this subject has a good PPV despite the sensitivity being poor, and

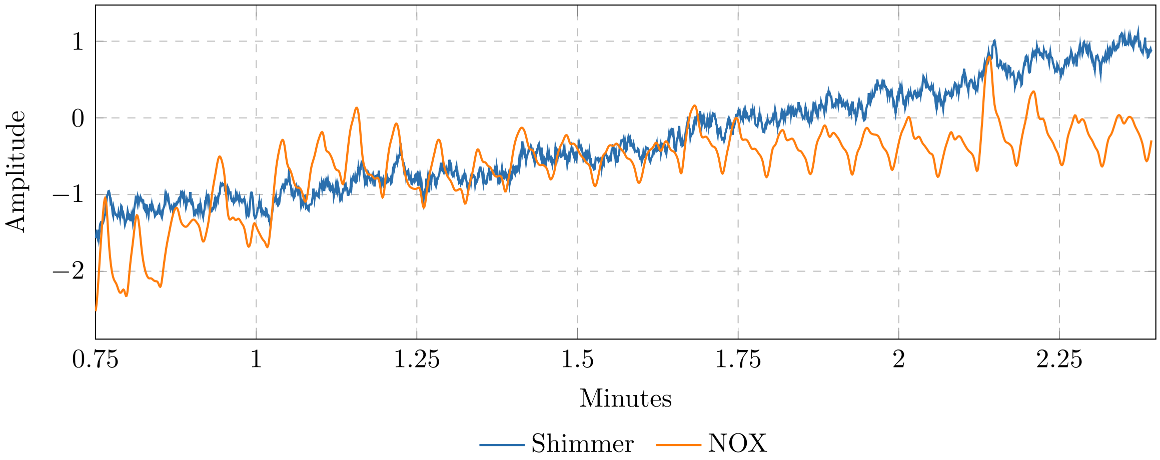


Figure 7.18: Example of a corrupt Shimmer capture

additionally, the signal from the supine position shows an imperfect PPV despite the sensitivity being perfect. In contrast to sensitivity, it is the side position that shows the best PPV on average with a score of 97.55%, where it is 96.58% for the supine position. This is, however, only when the corrupt signals are excluded, and the picture would surely be different if they were included (e.g., by regarding their score as 0% or something similar).

The CMP results (Figure 7.17c) show that a total of seven out of 24 of the signals maintain a perfect breath detection accuracy throughout the full duration of the capture. The supine position signal from Subject 1, 3, 5, and 7, and the side position signal from Subject 5–7, all show a perfect breath detection accuracy. The results show that both missing and false breaths are somewhat evenly distributed throughout the signals. The worse the sensitivity, PPV, or both, the worse the CMP, which corresponds to an even distribution.

There is possibly a trend related to BMI in the CMP results for the side position. Based on these results, the signal quality is best for the subjects with a BMI close to 25, and decreases as the distance to a BMI of 25 increases, in both directions. Since CMP depends on both the sensitivity and PPV, this trend is also present in the results for these metrics as well, although to a slightly lesser degree.

7.4.3 Breath Amplitude Accuracy

In contrast to the breath amplitude accuracy of the abdominal and sum-signals from BITalino, Shimmer shows more variation amongst the subjects. Both the supine and side position signals vary a lot amongst all the subjects, even between those with a very similar BMI. The only slight stability in amplitude accuracy regarding BMI present in the data is the supine signal between Subject 10–12 and the side signal between Subject 4–7. The amplitude accuracy for both positions varies by as much as 25% between subjects, and the mean amplitude accuracy is 16.89% and 21.37% for the supine and side positions, respectively. As a result, the signal from the supine position yields the overall best breath amplitude accuracy, which is also in line with the results for BITalino. Despite the side position signal being worse in general, there is a correlation between the supine and side position scores (at least for Subject 2–7 and 11). The worse the accuracy of the supine position, the worse the accuracy of the side position, and vice versa.

The best-achieved breath amplitude accuracy is 6.93% from Subject 4 in the supine position. Both signals from Subject 7 are, in fact, the only other two signals from Shimmer with a score below 10% (8.92% and 8.74% for supine and side, respectively). Figure 7.19 shows a scatterplot of the breath amplitude relationship between Shimmer and NOX, including the regression line, for a few selected signals. The selection is again based on all the different kinds of relationships present in the data and shows how such relationships affect quality. The sub-figures are sorted based on quality, with Figure 7.19a, Figure 7.19b, Figure 7.19c, and Figure 7.19d having a score of 6,93%, 12.14%, 17.17%, and 31,05%, respectively. Whereas most of the breaths are very close to the regression line in Figure 7.19a, hence the good score, the deepest breaths hint at a more monotonic relationship. On the other hand, as there are few deep breaths, this may very well also just be scatter as opposed to a different type of relationship altogether. The relationship of Figure 7.19b is very linear, but it also has more variation, which is what degrades the quality. The deep breaths (excluding the two outliers) of Figure 7.19c may indicate a slightly monotonic relationship, but it is hard to determine definitely because of the large variation. Despite the fact that Figure 7.19d shows a very linear relationship with minor scatter for the shallow and normal breaths, the overall relationship is very monotonic because of the deep breaths. The poor breath amplitude accuracy score (31,05%) of the signal is due to the large amplitudes of the deep breaths in combination with the monotonic relationship. Another common type of monotonic relationship present in the data is where the slope of the deep breaths increases, as opposed to decreases as it does in Figure 7.19d. The breath amplitude relationships from Shimmer are, in other words, never consistent amongst different subjects and across signal captures.

7.4.4 Conclusions

Regarding the breath detection accuracy, we arrive at mostly the same conclusion for Shimmer as for BITalino. However, while the main concern for BITalino is mostly the presence of false breaths, Shimmer struggles with both missing and false breaths. For Shimmer, the issue with sensitivity primarily concerns the higher end of the BMI scale for the supine position, while it concerns both ends of the BMI scale for the side position.

The raw signal from Shimmer is never a nearly flatline due to the inflicted high-frequency current. Regardless, this high-frequency current is very systematic and is, thus, also trivial to identify and filter out. The main withstanding issue is then the heartbeats. If the scoring rules by the AASM are followed strictly, Shimmer performs very poorly at detecting apneas because heartbeats are very prominent features of the signal; and often above 10% of the baseline in amplitude. One would either have to carefully filter the heartbeats out of the signal or identify and take them into consideration when scoring apneic epochs.

Whenever the signal-to-noise ratio is low, the shallow breaths have a tendency to be completely buried in noise. The result is that epochs which in reality are hypopneic may be misclassified as apneic. In fact, the shallow breaths are for all the signals from both

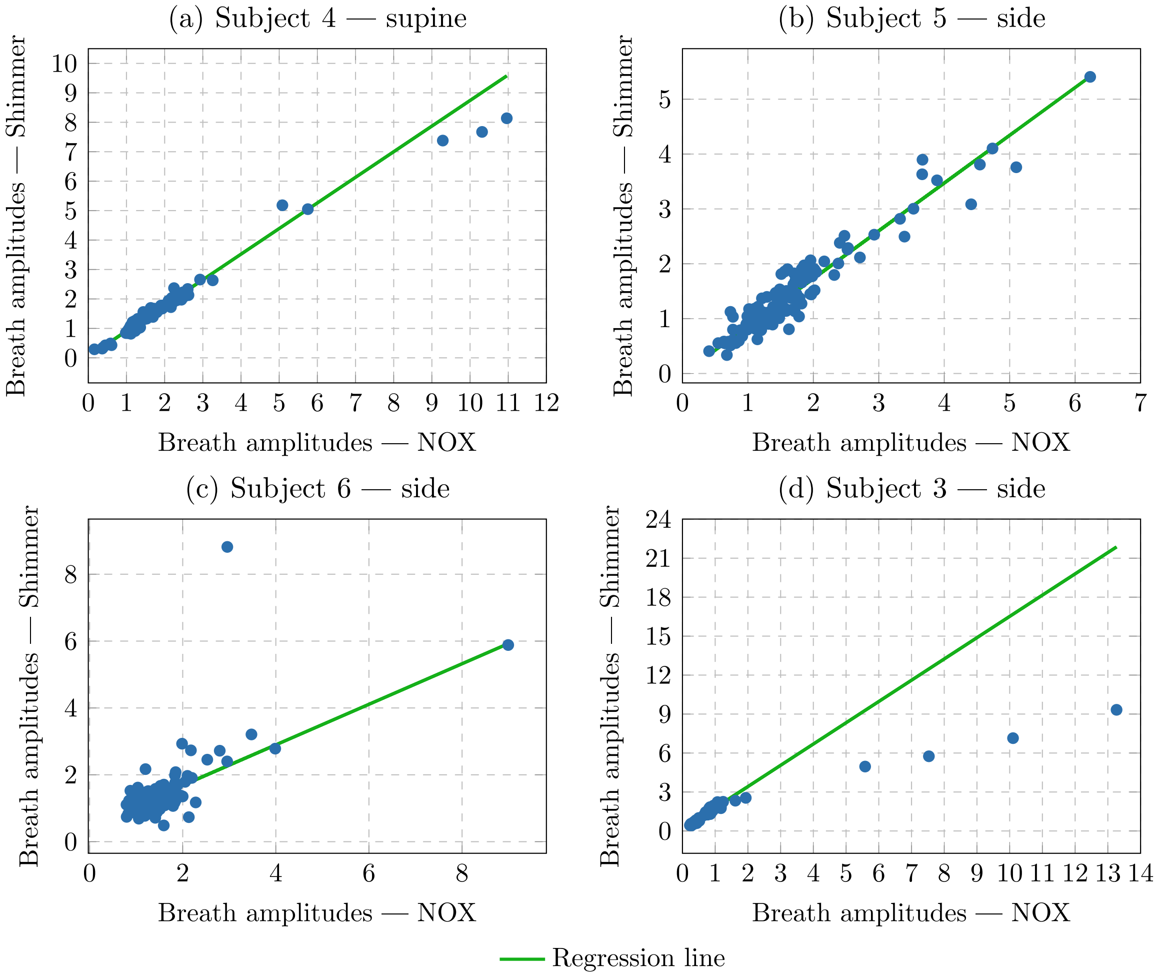


Figure 7.19: Breath amplitude relationship between Shimmer and NOX

positions, *always* either at or below the regression line, never above. This means that either hypopneic episodes may be misclassified as apneic or, if heartbeats are not taken into consideration, apneic episodes may be misclassified as hypopneic.

There is an issue regarding BMI affecting the signal from Shimmer as the signal quality is for the most part superior for the subjects with a BMI close to 25, for both positions. The issue is most prominent for the side position signal, and its presence in the supine signal is somewhat questionable. Despite the CMP score, the results for the supine signal from Subject 12 are in fact very good. Moreover, the CMP score for this signal is explained by a few false/missed breaths very evenly spread throughout the signal.

7.5 Platform Comparison

Figure 7.20 presents a comparison of all the metrics for the overall best signals from BITalino and Shimmer. The signal from BITalino is the raw abdominal signal from the supine position, and the Shimmer signal is from the supine position as well.

From Figure 7.20a, it becomes clear that BITalino has the better sensitivity. Excluding the subjects where both sensors have a sensitivity of 100% (Subject 1–5), BITalino is better in five out of seven cases, whereas Shimmer is better in the other two. The mean sensitivity of BITalino and Shimmer is 99.61% and 98.53%, respectively. For the PPV (Figure 7.20b), the results are quite different. While only one out of twelve signals from BITalino shows a perfect PPV, four out of eleven signals from Shimmer do. As BMI increases, both sensors struggle with false breaths, with Shimmer struggling slightly more than BITalino (Subject 10–11). The mean PPV of BITalino and Shimmer is 96.28% and 96.58%, respectively, making the average score of Shimmer only marginally better. The corrupt signal from Shimmer (Subject 8) is not taken into account for the calculation of this mean, and the results would surely be different if it were. It is already determined that the distribution of missed/false breaths are for both sensors very even, and so the CMP metric (Figure 7.20c) is less interesting to compare. One may, however, notice Subject 6 which has a lower sensitivity and PPV for Shimmer, but simultaneously also a better CMP. This observation indicates that the errors are more spread throughout the signal for BITalino; for this specific subject.

Based on the breath amplitude accuracy results shown in Figure 7.20d, it is apparent that Shimmer shows the most variation amongst subjects. BITalino shows somewhat stable results between Subject 2–5, whereas Shimmer does between Subject 10–12. Regarding BMI, BITalino does indicate a correlation as the results are much less stable between subjects at both ends of the scale. For Shimmer, on the other hand, the correlation is very questionable for this specific signal and metric. The mean breath amplitude accuracy of BITalino and Shimmer is 13.82% and 16.89%, respectively, making BITalino the overall most accurate of the two.

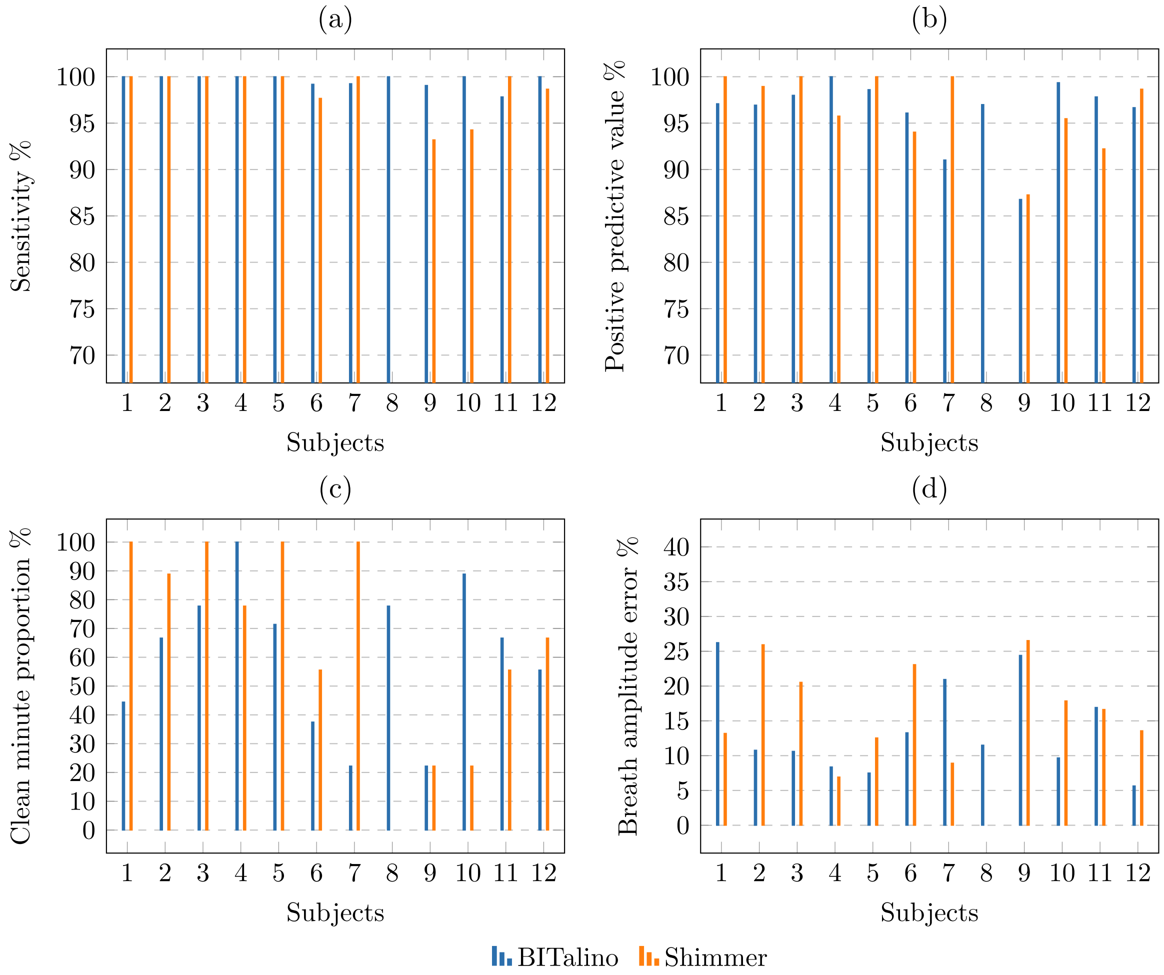


Figure 7.20: Comparison of all quality metrics for the supine signal from BITalino (raw, abdomen) and Shimmer

7.6 Metric Review

Breath Detection Accuracy

Both sensitivity and PPV are straightforward metrics which are very easy to interpret, and also widely used amongst practitioners to describe the performance of binary classifiers. When these metrics are used to describe the performance of automatic binary classifiers, the results reflect the objective truth. Our case is, however, a little different. The sensors provide an analog signal, and we extract the breaths based on our subjective evaluation. If we, for example, blindly trusted our automatic breath detection implementation, the results would be objective, but the metrics would then describe the performance of the extraction algorithm rather than the signals alone. That said, the accuracy of the algorithm is almost flawless whenever the signal is as clean as it is for the NOX sensor but struggles with more noisy signals. There is nothing wrong with the metrics per se, but it is, however, worth noting the subjective aspect of their use in this context. A solution to reduce the subjectivity could, for example, be to have multiple (trained) persons score the signals, but unfortunately, we do not have the time nor resources to do so during this work.

Of all the three breath detection accuracy metrics, CMP yields the least information, and its usefulness is questionable. We do realize that a window width of one minute may be too large considering that the signal captures only consist of about seven minutes. A window width of about 15–30 seconds may have been more optimal for these signal captures, but regardless, the metric is more fit for longer sessions anyhow.

Breath Amplitude Accuracy

The standard definition of the WAPE metric (Equation 7.1) consists of two arithmetic means, the denominator ($\overline{y}$) and the mean of all the individual errors (the $\frac{1}{n}\sum_{i=1}^{n}$ term).

$$WAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{\overline{y}}\right| \times 100\% \tag{7.1}$$

In the context of breath amplitude accuracy, the denominator ($\overline{y}$) represents the *baseline breath amplitude*. Whether or not the arithmetic mean is the best representation of such, depends upon the underlying distribution of the breath amplitudes. The breath amplitude distribution for a few selected raw abdominal BITalino signals are shown in the left column of Figure 7.21 (Figure 7.21a, Figure 7.21c, Figure 7.21e, and Figure 7.21g), and for reference, these signals are the same as those shown in Figure 7.16. As seen, the distribution of the breath amplitudes is not perfectly normal but more right-skewed, and the degree of the skewness varies between the signals. For skewed distributions in general, the *median* is often considered a better representation of central tendency compared to the arithmetic mean. For these signals, the median and mean are fairly close (and even overlapping in Figure 7.21a), but the median is consistently slightly smaller. Choosing the median in place of the mean as the denominator for the WAPE metric, therefore, systematically increases the error (a smaller denominator equals a larger error).

As seen in the right column of Figure 7.21 (Figure 7.21b, Figure 7.21d, Figure 7.21f, and Figure 7.21h), the distribution of the breath amplitude errors is very right-skewed, and the difference between the median and mean is significant (pay attention to the axes values). Due to the skewness of the distribution, choosing the median as the measure of central tendency minimizes the effect of outliers, but also systematically decreases the average amplitude error. The effect of choosing the median in place of the mean for these signals can be seen in Figure 7.22. This figure presents all the different combinations of mean and median for the raw abdominal supine signal from BITalino and the supine signal from Shimmer. The legend notation *median of mean* means that the outer term $\frac{1}{n}\sum_{i=1}^{n}$ is converted to median, while the inner term (denominator) $\overline{y}$ still represents the arithmetic mean. The results show that converting the outer term to median has the greatest impact on the metric score, and the impact of outliers is also clearly visible. While most trends are still present, the variation between subjects is minimized significantly.

In conclusion, which one of these definitions that is most optimal for these kinds of data, depends on what the metric should represent. If the intention is to minimize the effect of outliers on the metric score, then the median is superior, and if not, then the standard definition is superior. Interestingly, the use of median for these kinds of data is very rarely (if at all) mentioned in related work. Most related work employs metrics which utilize the arithmetic mean in one way or another, such as MAE, MAPE, RMSE, and MBE ((Silva et al. 2015), (Seppänen et al. 2013), (Liu et al. 2013), (Cantineau et al. 1992), (Adams et al. 1993), and (Cohn et al. 1982)). Nonetheless, the use of median remains a very viable alternative for these kinds of data.

7.7 Comparison with Related Work

Brouillette et al. (1987) evaluate the breath detection accuracy of an IP sensor (what Shimmer uses) and a RIP sensor (what we use with NOX) using sensitivity and PPV as the metrics. In their study, the IP sensor shows a sensitivity of 98.3% and a PPV of 94.4%. Conversely, their RIP sensor shows a sensitivity of 99.6% and a PPV of 99.5%. Both sensor types identify 60 out of 60 central apneic events, but for obstructive events, on the other hand, the picture is quite different. The RIP sensor identifies 35 out of 38 obstructive events, whereas the IP sensor identifies only two out of 38. The study shows that false breaths caused by cardiac activity and respiratory effort are very significant issues concerning the IP type sensor.

Our results for Shimmer are in line with the results from this study. Even considering the subjective aspect of the breath detection metrics, their finding regarding sensitivity is very similar to our results for Shimmer from the supine position (98.3% versus 98.5%). The results for PPV are less similar (94.4% versus 96.5%), but their result includes two large outliers with a PPV of 56% and 46%, which may explain the difference. Additionally, the study substantiates our findings regarding false breaths and heartbeats being very significant concerns regarding this kind of sensor.

Concerning breath amplitude accuracy, a direct comparison with related work is not possible as the metrics in use are different. For example, (Adams et al. 1993), (Cantineau et al. 1992), (Whyte et al. 1991), and (Cohn et al. 1982) all measure the breath amplitude accuracy of different sensor types, but employ variations of either MPE or MBE as the accuracy metric. What we can compare, however, is the consistency of the breath

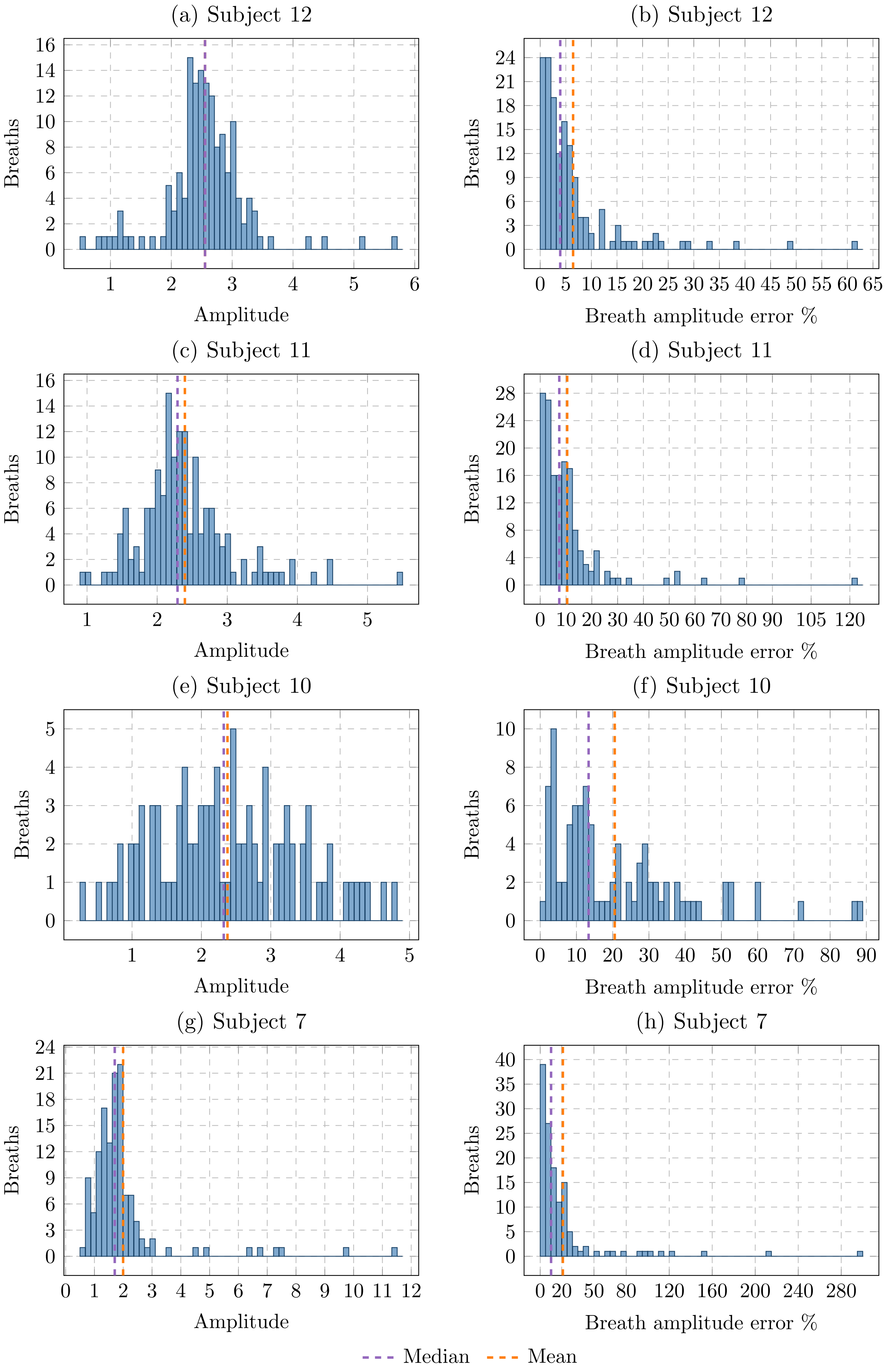


Figure 7.21: Distribution of breath amplitudes (left) and breath amplitude errors (right)

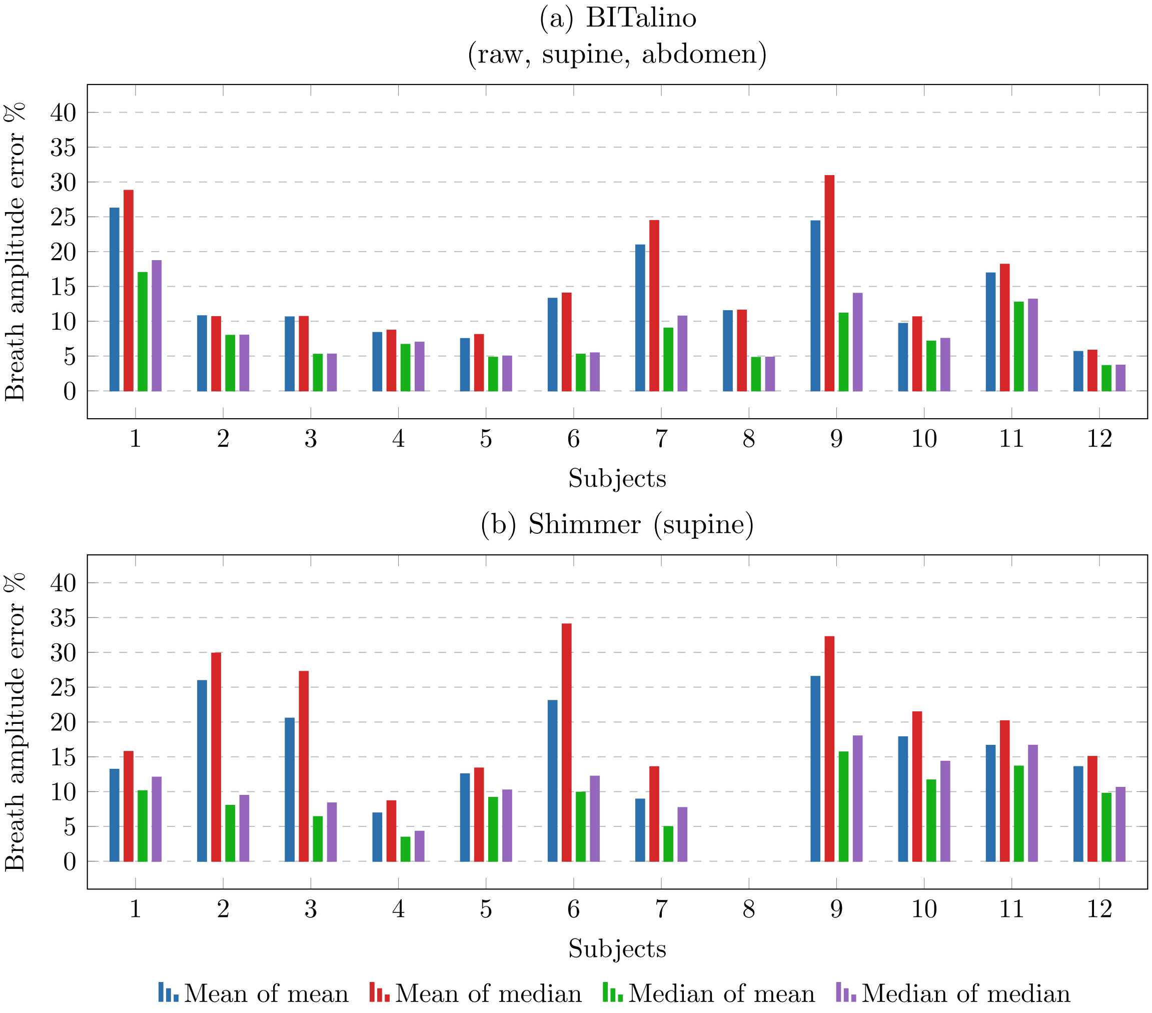


Figure 7.22: Comparison of the different combinations of median and mean for the WAPE metric

amplitude relationship across subjects. Our results show that the amplitude of the breaths are not consistently higher or lower than the gold standard across subjects, but varies. This is in line with the results by Cantineau et al. (1992), who measure the accuracy of a RIP type sensor.

7.8 Additional Sensors

Towards the very end of this work, we received two additional respiratory effort sensors for evaluation: a RIP belt from *biosignalsplux* called *RespiBAN* (biosignalsplux 2018b), and a strain-gauge belt from *SweetZpot* called *FLOW* (SweetZpot 2018). Both of these sensors are single belts, meaning that they capture either abdominal *or* thoracic movement. Given the limited time frame and the (usually) superior signal quality from the abdomen, we capture only abdominal movement during the evaluation of these two sensors. For this evaluation, we regathered a subset of the original subjects as well as some newcomers, resulting in a total of eleven subjects.

7.8.1 RespiBAN

An example capture from RespiBAN is shown in Figure 7.23. Notice how similar the waveforms of the signal are to the airflow signal from the BITalino PZT belts. As soon as airflow subsides, without the belt circumference changing, the signal of the RespiBAN sensor immediately returns to the baseline center. This behavior is just as the BITalino PZT belts and is precisely how an airflow signal should behave. However, the RespiBAN is a RIP type sensor, and *plethysmography* is defined as the *change of volume* (OED 2018b). Whether this sensor captures airflow or volume is, therefore, unclear, and it may very well also be some internal filter at work causing this behavior. Nonetheless, we tried to integrate the raw signal, but the results are the same as for BITalino, i.e., slightly worse overall than the raw signal.

All the signal quality metrics for the RespiBAN sensor are presented in Figure 7.24. The sensitivity is overall very good for both body positions with a score above 95% for all signals (Figure 7.24a). In contrast to both BITalino and Shimmer, the sensitivity of the side position for this sensor is most often better than the supine position. The mean sensitivity of the supine and side positions is 98.41% and 98.88%, respectively.

As shown in Figure 7.24b, the PPV of the RespiBAN sensor is somewhat poor. The mean PPV of the supine and side positions is 90.81% and 86.64%, respectively, making it achieve the worst PPV of all the sensors. The underlying cause is not related to sporadic false breaths throughout the signal, but rather the fact that the signal cannot flatline at all. During the breathing stops, the signal exhibits behavior that is often misinterpreted as shallow breaths, which in turn results in a poor PPV score. During a breathing stop, BITalino, Shimmer, and FLOW do not flatline either, but their behavior is at least mostly distinguishable from shallow breaths. The breathing stops in these signal captures are often located across two minutes of the signal (i.e., stretching across the boundary of one minute onto another). Therefore, the two breathing stops may be located within four minutes, which explains why the CMP score of this sensor is often around 50% (Figure 7.24c).

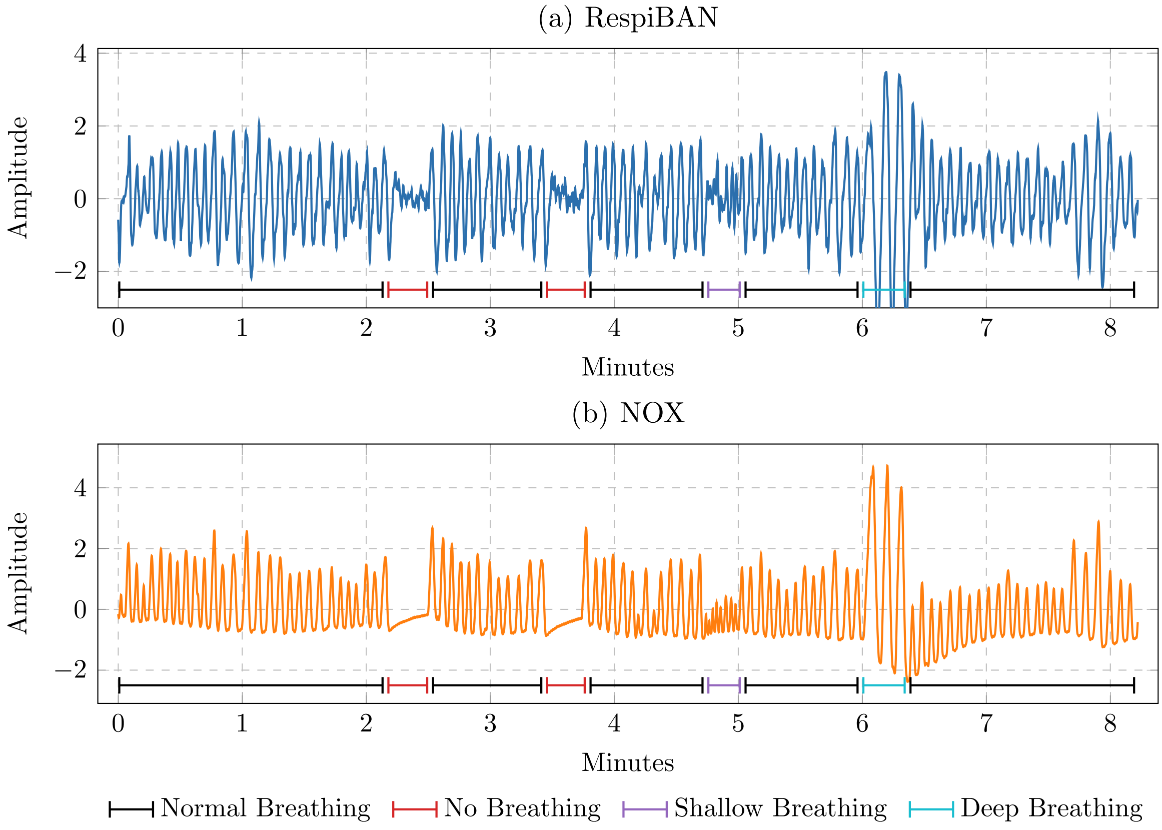


Figure 7.23: Example of a RespiBAN capture

The breath amplitude accuracy, on the other hand, is reasonably good overall with the mean score for the supine and side positions being 13.60% and 14.65%, respectively (Figure 7.24d). The score of all signals is remarkably stable across captures and subjects, and usually reside in the 10–15% range. In contrast BITalino and Shimmer, this sensor shows only a single outlier (Subject 2 side) in regards to breath amplitude accuracy.

7.8.2 FLOW

An example capture from the FLOW sensor is shown in Figure 7.25. The first point one may notice is that this sensor does suffer from a slight baseline wander. It is not so severe that it affects the synchronization in these captures, but it might be for longer captures. Baseline wander is, nonetheless, easily correctable. It is clear that this sensor captures volume as the signal does not return to the baseline center when airflow subsides. The signal is, on the other hand, quite noisy (evident during breathing stops). However, the noise is of somewhat high frequency, making it easily distinguishable from breaths in general.

All the signal quality metrics for FLOW are shown in Figure 7.26. The sensitivity of the FLOW sensor is also very good (Figure 7.26a), with the mean scores of the supine and side positions being 98.91% and 98.22%, respectively. Unlike the RespiBAN sensor, FLOW exhibits very few false breaths (Figure 7.26b). The mean PPV of the supine and side positions is 98.81% and 99.16%, respectively, making it the best sensor regarding PPV score. The signal from FLOW is one of the more noisy signals amongst these

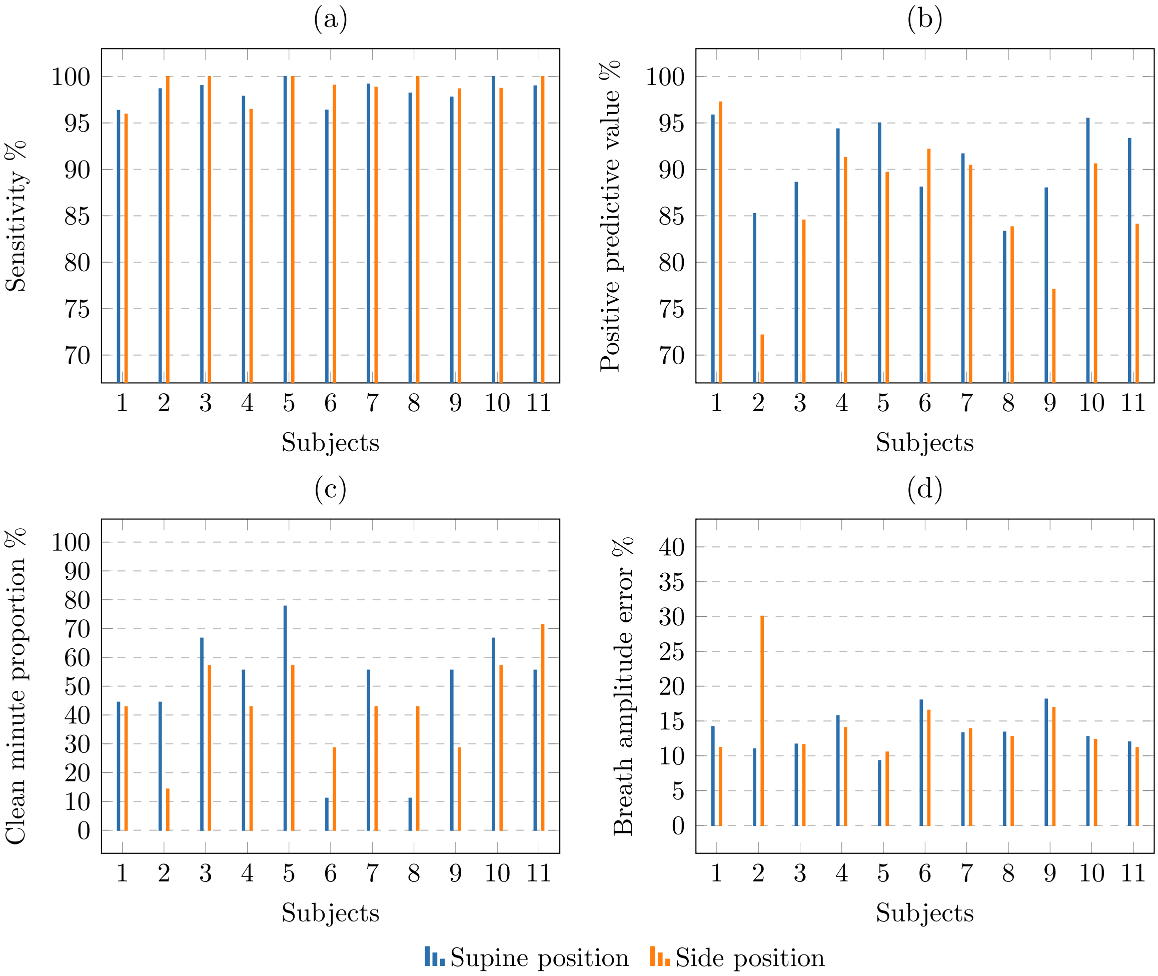


Figure 7.24: All quality metrics for all signals from RespiBAN

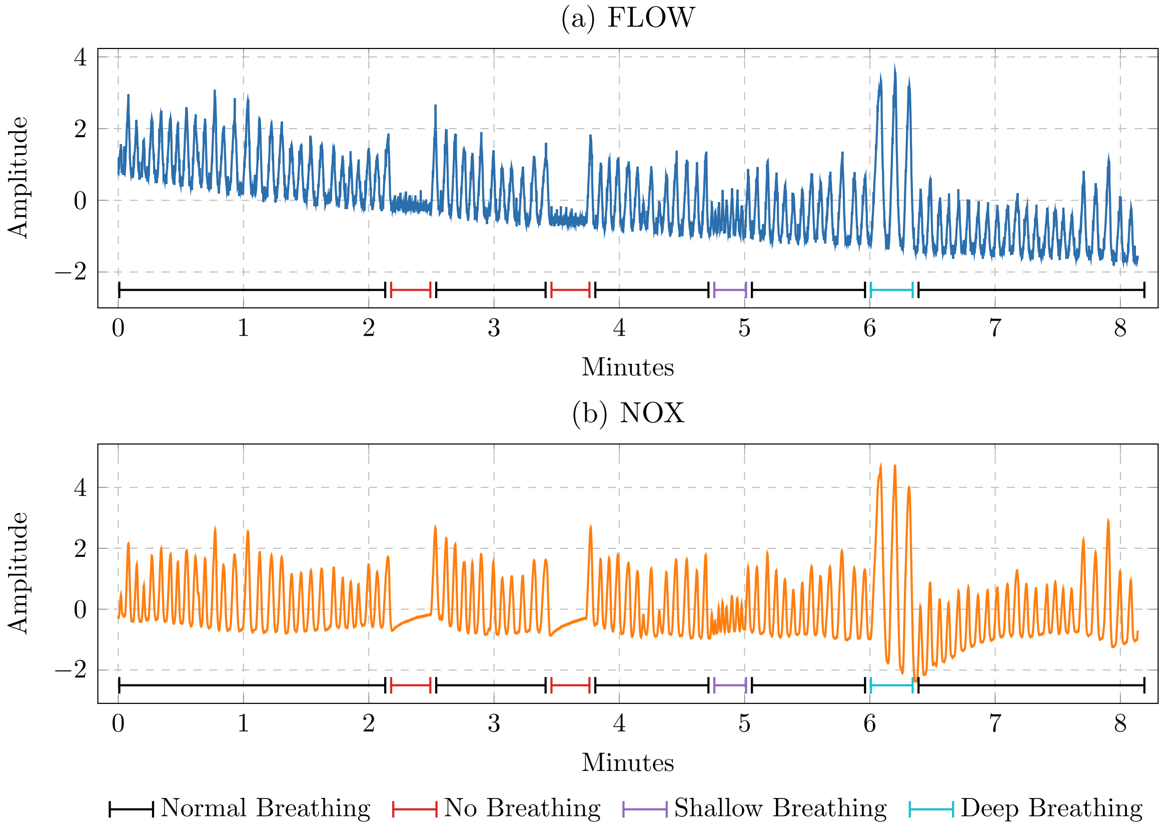


Figure 7.25: Example of a FLOW capture

sensors, but its noise is also easily distinguishable from breaths in general. With such a good sensitivity and PPV scores, there is not much to mention regarding the CMP (Figure 7.26c). Although, the CMP scores indicate that the few errors present in the signals are somewhat evenly distributed throughout the signals (i.e., CMP is poor despite sensitivity and PPV being good).

FLOW achieves the best breath amplitude accuracy of all the sensors (Figure 7.26d). The score is centered below 10% and is very stable with few to no outliers. The mean score for the supine and side positions is 8.75% and 9.61%, respectively. The reason why the FLOW sensor achieves this much better breath amplitude accuracy compared to the other sensors is unclear. One possible explanation is that it is the only belt type sensor which captures the same unit of measurement as the NOX (volume), making them very closely related. Of all the other sensors, it is only Shimmer which *definitively* captures volume, but Shimmer is also a very different type of sensor compared to NOX.

7.9 Discussion and Conclusions

Based on the signal captures from twelve subjects, we conclude that false breaths are the primary concern affecting the breath detection accuracy of both BITalino and Shimmer, where Shimmer is also somewhat struggling with missing breaths. Both platforms show an inconsistent relationship between breath amplitude and changes in body circumference (due to breathing), with varying types of monotonic relationships being the norm. The

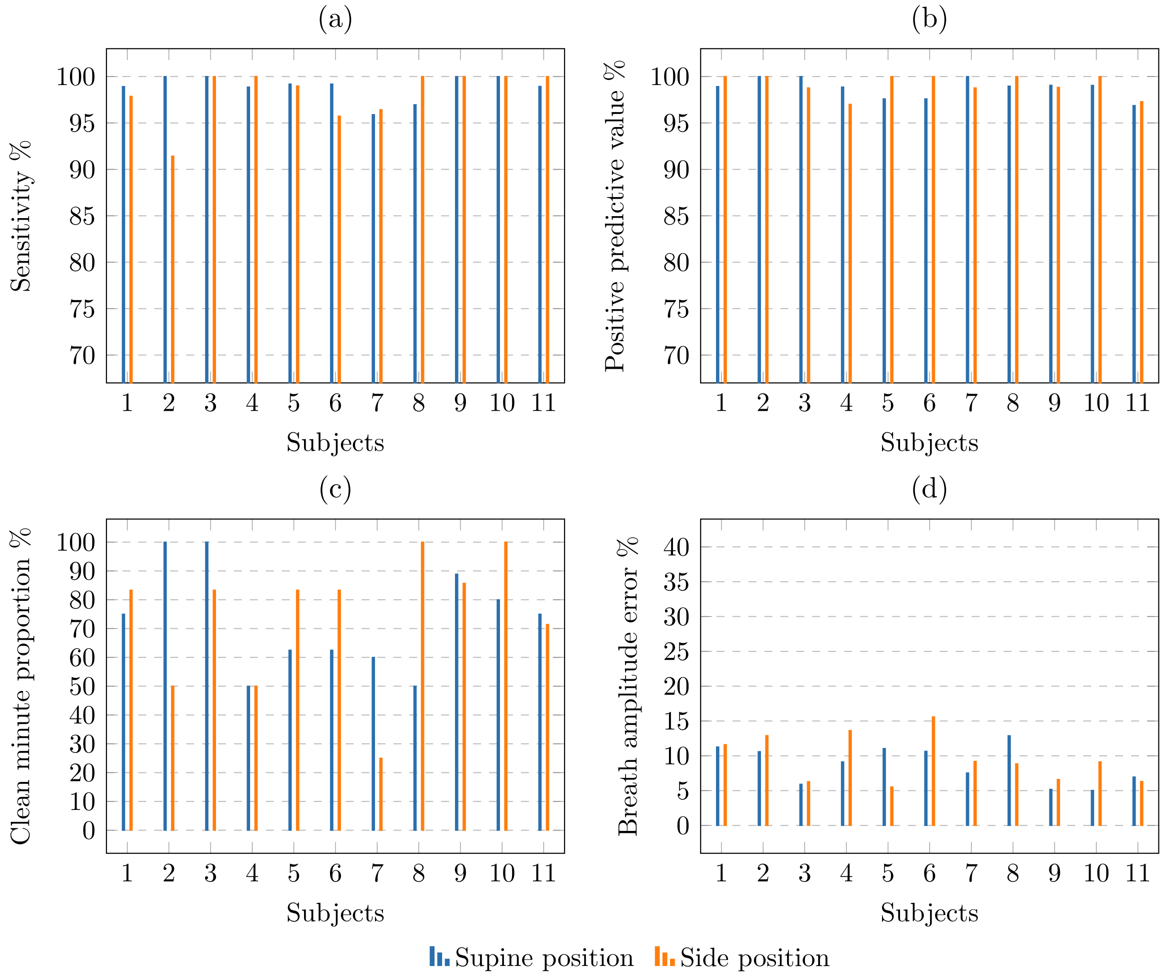


Figure 7.26: All quality metrics for all signals from FLOW

supine body position is consistently showing the overall best signal quality. While both platforms show a correlation between signal quality and BMI, the supine position is less affected than the side position.

The RespiBAN sensor is mostly struggling with false breaths during breathing stops. Its sensitivity and breath amplitude accuracy are very close to the raw BITalino signals, but the RespiBAN signal is more stable across body positions and signal captures. Despite being one of the noisiest signals, the FLOW sensor achieves overall the best metric scores. In fact, only the sensitivity of the raw abdominal BITalino signal beats it. The FLOW sensor is also, like RespiBAN, very stable across body positions and signal captures. The more stable results between body positions for the RespiBAN and FLOW sensors is suspected to be related to sensor entrapment. The sensor part of the RespiBAN and maybe FLOW (documentation lacking) stretches the whole circumference of the belts, whereas only a small area for BITalino. Furthermore, the Shimmer sensor is likely affected by lying directly on top of the electrode attached to the side of the thorax. An overview of the mean metric score for each signal and sensor is shown in Table 7.1, with the best signal for each metric emphasized. Figure 7.27 shows an example of what a complete breathing stop looks like for all the different sensors. The left column is the target sensors, whereas the right column their corresponding gold standard signal from NOX.

It is important to note the subjective aspect of the sensitivity and PPV metrics when they are applied to this context. Without a ”perfect” automatic breath extraction algorithm, this subjective aspect is unavoidable, but it can be minimized by, for example, having multiple (trained) persons manually score the signals. The CMP metric is the least useful of these three breath detection accuracy metrics, and when the signals under evaluation are very short, the window width of one minute may be too wide.

Depending on whether or not the effect of outliers should be minimized for the breath amplitude accuracy metric, the median is a very viable alternative to the arithmetic mean. The distribution of the breath amplitude errors is, for these signals, right-skewed, and the median is often considered a superior representation of central tendency for skewed distributions. In the end, the choice depends upon what we intend the metric to describe.

	Sensitivity	**PPV**	**CMP**	**WAPE**	**Corrupt**
BITalino (raw, abdomen, supine)	**99.61%**	96.28%	60.93%	13.82%	0 of 12
BITalino (raw, abdomen, side)	99.16%	93.83%	53.24%	16.51%	1 of 12
BITalino (raw, thorax, supine)	97.47%	94.69%	41.00%	20.60%	0 of 12
BITalino (raw, thorax, side)	97.81%	92.24%	47.14%	22.36%	2 of 12
BITalino (raw, sum, supine)	99.48%	96.44%	67.98%	14.28%	0 of 12
BITalino (raw, sum, side)	99.29%	94.71%	64.93%	16.51%	1 of 12
BITalino (integrated, abdomen, supine)	95.79%	96.74%	66.98%	19.36%	0 of 12
BITalino (integrated, abdomen, side)	96.06%	96.60%	50.64%	21.85%	1 of 12
BITalino (integrated, thorax, supine)	93.18%	98.43%	51.33%	28.42%	0 of 12
BITalino (integrated, thorax, side)	90.62%	95.60	42.85	36.31%	2 of 12
BITalino (integrated, sum, supine)	96.48%	98.46%	66.48%	19.27%	0 of 12
BITalino (integrated, sum, side)	96.80%	97.41%	57.14%	22.60%	1 of 12
Shimmer (supine)	98.53%	96.58%	71.72%	16.89%	1 of 12
Shimmer (side)	97.30%	97.55%	70.11%	21.37%	3 of 12
RespiBAN (supine)	98.41%	90.81%	49.50%	13.60%	0 of 11
RespiBAN (side)	98.88%	86.64%	44.16%	14.65%	0 of 11
FLOW (supine)	98.91%	98.81%	73.08%	**8.75%**	0 of 11
FLOW (side)	98.22%	**99.16%**	**74.13%**	9.61%	0 of 11

Table 7.1: Overview of the mean metric scores of all signals

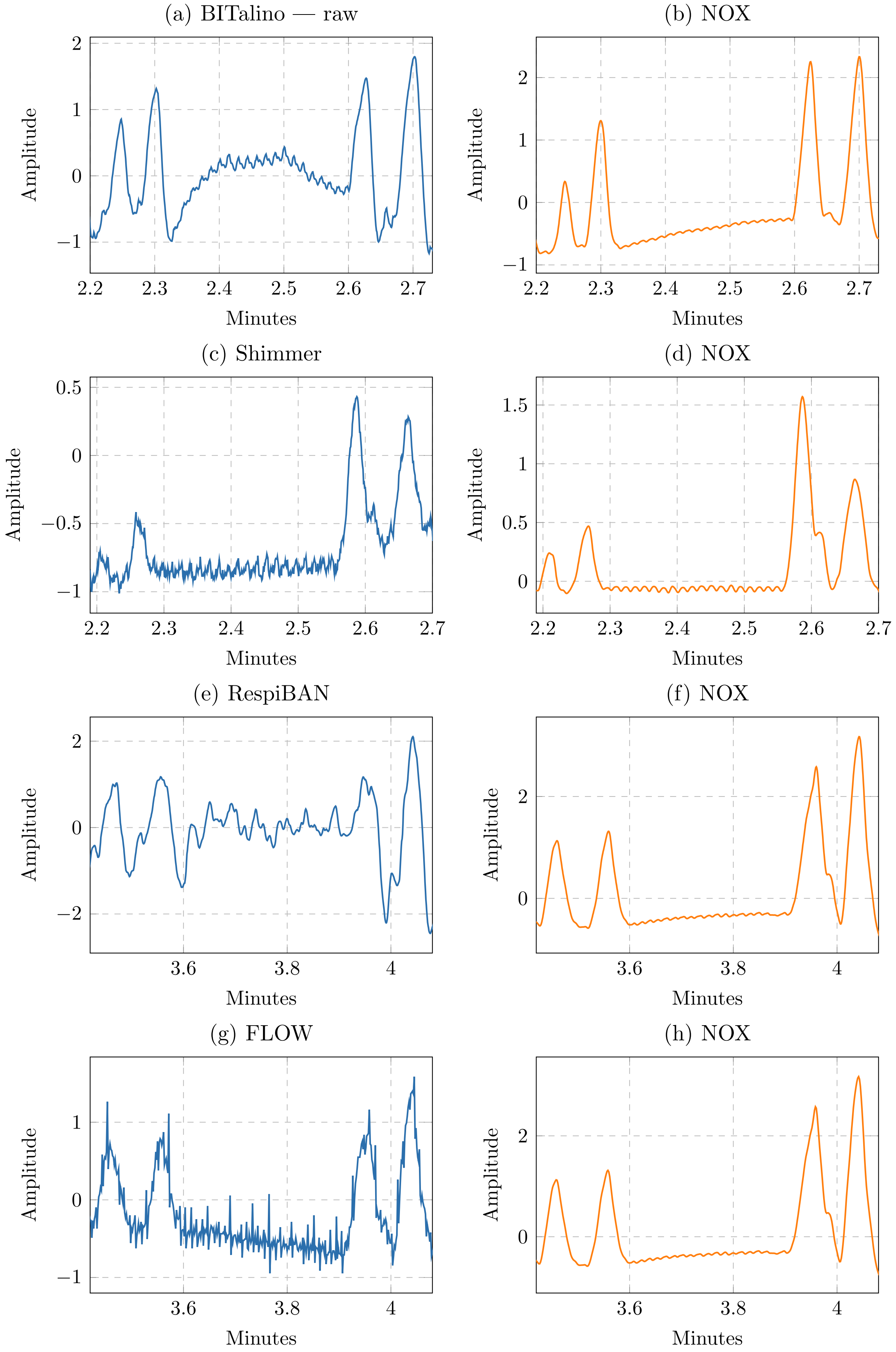


Figure 7.27: Example of a complete breathing stop from all sensors

Chapter 8

Conclusion

This chapter concludes the work of this thesis. We begin in Section 8.1 by presenting a summary of the main contributions made in this work. This includes the signal capture procedure, the metrics, and the results of the four sensors, BITalino, Shimmer, RespiBAN, and FLOW. We continue in Section 8.2 with a critical assessment of the work and methods, before we end the thesis by presenting a few possible directions for future work in Section 8.3.

8.1 Summary of Contributions

In this work, we evaluate the signal quality of four respiratory effort sensors for sleep apnea monitoring. Namely a *piezoelectric effort belt* (PZT) from BITalino, an *impedance plethysmography* (IP) sensor from Shimmer, a *respiratory inductance plethysmography* (RIP) sensor (RespiBAN) from biosignalsplux, and a strain-gauge sensor (FLOW) from SweetZpot. We use a RIP sensor from NOX Medical as the gold standard. To evaluate the signal quality of these sensors, we design a sixteen-minute signal capture procedure and capture data from twelve (BITalino and Shimmer) and eleven (RespiBAN and FLOW) external subjects. Our signal quality evaluation approach is based on the breath detection accuracy metrics *sensitivity*, *positive predictive value* (PPV), and *clean minute proportion* (CMP), along with the breath amplitude accuracy metric *weighted absolute percentage error* (WAPE). From BITalino, we measure the signal quality of 48 raw signals (24 abdominal and 24 thoracic signals), along with 96 logical signals (i.e., sum-signals and integrated counterparts). From Shimmer, we measure the signal quality of 24 raw signals. Additionally, we also measure the signal quality of 22 signals each from RespiBAN and FLOW. In total, the quality of 212 different signals is evaluated in this work.

8.1.1 Signal Capture Procedure

The direct method to measure the signal quality of a sensor for sleep apnea monitoring is to include it in traditional polysomnography using real sleep apnea sufferers, and then manually score the results. This is, however, very resource demanding and time-consuming. Given our limited set of resources and time frame for this thesis, we instead design a shorter signal capture procedure that can be performed in a laboratory during wakefulness. The full duration of the procedure is sixteen minutes, and it simulates disrupted breathing through shorter periods of shallow, deep, and no breathing. The complete signal capture procedure is defined in Section 5.3.3.

The signal capture procedure is generic and not specifically designed for the respiratory effort sensors we are evaluating. It is designed for any sensor that directly monitors the respiratory process, such as an *oronasal thermal sensor*, *nasal pressure transducer*, and most types of respiratory effort sensors. The procedure may be used with sensors that indirectly monitor the respiratory process as well, such as a pulse oximeter or ECG, but the duration of the periods of disrupted breathing may need to be adjusted. For example, a period of 10–20 seconds may not be long enough to affect the SpO_2 levels as much as necessary.

8.1.2 Metrics

Many related studies evaluate the signal quality of respiratory sensors based on either the signal as a whole or the accuracy of each breath in isolation. The result is that many aspects of the signals that are irrelevant in the context of sleep apnea monitoring are still included in the signal quality evaluation. We instead employ metrics which are closely related to how medical personnel scores apneic and hypopneic episodes. Apneic episodes are ultimately scored based on the absence of breaths. It is, therefore, only false and missing breaths that affect a sensor's ability to detect apneic events. The *sensitivity* metric directly reflects the proportion of missing breaths, and the *positive predictive value* metric directly reflects the proportion of false breaths. However, if the presence of false/missing breaths occurs seldom, but in bursts, these two metrics may still be significantly affected even though most parts of the signal are very accurate. As a result, we also employ the *clean minute proportion* metric, which reflects the proportion of minutes in the signal that are 100% accurate (i.e., both sensitivity and PPV are 100% during the minute). Hypopneic events are scored based on a 30% reduction in breath amplitude relative to the *baseline* breath amplitude. The *weighted absolute percentage error* metric describes precisely this, the accuracy of the breath amplitudes in relation to the baseline breath amplitude.

8.1.3 Signal Quality of the Target Sensors

BITalino and Shimmer

Based on the signal data captured from twelve external subjects, we evaluate the signal quality of a piezoelectric belt from BITalino (both abdominal and thoracic) and an impedance plethysmography sensor from Shimmer. The PZT sensor from BITalino captures *airflow*, whereas the gold standard (NOX RIP belts) and the Shimmer IP sensor capture *volume*. As of this fact, one would expect that the integrated signal from BITalino should be closer to the gold standard signal compared to the raw signal. However, our results show that the raw signals are superior in most cases. We suspect that the underlying cause includes noise and measurement errors regarding breath amplitude and duration, which are amplified by the integration.

We evaluate the signal quality of both the BITalino and Shimmer sensors from two different sleeping positions, the supine (back) and the side position. For both sensors, the supine position shows the superior results. For the BITalino sensor, it is the raw abdominal signal from the supine position that shows the best results overall. This signal shows on average a sensitivity, PPV, CMP, and WAPE score of 99.61%, 96.28%, 60.93%, 13.82%, respectively. Of all signal combinations, this signal achieves the best sensitivity metric score. Whereas the raw signals from BITalino achieve better scores overall compared to their integrated counterparts, the integrated versions achieve the best PPV scores. The reason is that integration acts as a low-pass filter, which effectively attenuates the false breaths from the signal. The primary signal quality concern for BITalino is the presence of false breaths. A large number of false breaths is expected to increase the rate of false negative apneic and hypopneic events. Of 48 raw signals (two from each subject from each body position), three are corrupt. Two thoracic signals and one abdominal signal, both from the side position.

As mentioned, the best performing signal from Shimmer is from the supine position as well. This signal achieves on average a sensitivity, PPV, CMP, and WAPE score of 98.53%, 96.58%, 71.72%, 16.89%, respectively. The signal quality of the Shimmer sensor is less stable between subjects compared to the BITalino sensor. In other words, the signal quality of Shimmer is often either very good or somewhat poor, but seldom in between. While the BITalino sensor struggles mainly with false breaths, the Shimmer sensor is also somewhat concerned with missing breaths. Out of 24 signals, four are corrupt. Three of those are from the side position of subjects with a BMI above 30, while one is from the supine position of a subject with an average BMI. There is a trend related to signal quality and BMI present in the data. The signal quality is worse on both ends of the BMI scale, and best close to a BMI of 25 (i.e., average).

RespiBAN and FLOW

The signal quality evaluation of RespiBAN and FLOW is based on data from eleven external subjects. From each of these subjects, we capture movement from the abdomen for both the supine and side body positions. Compared to BITalino and Shimmer, the signal quality from both the RespiBAN and FLOW is remarkably stable across different signal captures, subjects, and body positions. There are, in other words, far fewer outliers for these sensors. The RespiBAN sensor is severely struggling with false breaths during breathing stops because the signal just cannot flatline. The FLOW sensor is not struggling with anything in particular related to the signal quality metrics, but the signal is, nonetheless, very noisy. The supine signal from RespiBAN achieves on average a sensitivity, PPV, CMP, and WAPE metric score of 98.41%, 90.81%, 49.50%, and 13.60%, respectively. Likewise, the supine signal from FLOW achieves on average a sensitivity, PPV, CMP, and WAPE metric score of 98.91%, 98.81%, 73.08%, and 8.75%, respectively. Whereas lying in the side position reduces the occurrence of false breaths for the FLOW sensor, it increases it for the RespiBAN sensor.

8.2 Critical Assessment

The industry gold standard sensor for measuring airflow is a pneumotachograph (Berry et al. 2012). Due to the limited set of resources available for the work in this thesis, a pneumotachograph is unavailable, and we have to resort to other means. Therefore, we use a dual thoracobdominal RIP sensor from NOX Medical as the gold standard during this work. The RIP technology is shown to correlate very well with the signal from an integrated pneumotachograph. For example, Cohn et al. (1982) show that 88% of the tidal breaths recorded by RIP are within 10% the amplitude of the breaths recorded by a pneumotachograph. Additionally, the breath amplitudes recorded by a RIP sensor is shown not to be consistently lower or higher than the pneumotachograph, but vary. Consequently, this may potentially bias our results. For example, the RIP sensor (our gold standard) may record the amplitude of a breath as 10% *higher* than what a pneumotachograph *would* have, and the target sensor under evaluation may record the same breath's amplitude as 10% *lower* than what a pneumotachograph *would* have. This results in a 20% amplitude difference between the RIP sensor and the target sensor, while the real error is only 10%. As shown in our results, the breath amplitudes of both BITalino and Shimmer are also not consistently lower or higher than the RIP NOX sensor. It may, therefore, also be the case that the bias is canceled out and is insignificant, but this is nonetheless just speculations.

We must also emphasize that we are by no means medical personnel nor have any kind of medical training. While most of the breaths in the signals are unambiguous and trivial to score, some are very ambiguous. Consequently, a person with medical training *may* score the ambiguous breaths quite differently than how we do.

The design of the periods of disrupted breathing in the signal capture procedure is very "textbook" and "artificial/clean." For example, apneic and hypopneic events are in reality very likely to include paradoxical breathing, i.e., asynchronous breathing between the abdomen and thorax. As such, it may have been beneficial to study further what *real* apneic and hypopneic events look like, and then mimic this behavior in the signal capture procedure more closely. That said, the simpler variants in our procedure are hard enough to perform for the subjects as they are already. If we are to make them any more complicated, we expect that quite a few of the subjects would have a hard time performing them correctly.

8.3 Future Work

There are many opportunities for future work related to the topic of measuring the signal quality of respiratory effort sensors. First and foremost, consider the primary long-term goal of enabling people to perform the first step towards a diagnosis at home (Section 1.1). To tackle this goal, the relationship between the metrics we employ and the performance of the data mining classifiers must be studied. The results by Kristiansen et al. (2018) show that the abdominal signal achieves a better data mining classifier performance compared to the thoracic signal. These results are in line with our findings regarding the abdominal signal being of better quality compared to the thoracic signal. There is, however, too little data to conclude anything yet. If the results of the metrics we employ turn out to be highly correlated with the performance of the data mining classifiers, then our approach is a very efficient method to evaluate the signal quality of respiratory sensors in general.

We employ a minimal set of filtering techniques in this work. The signals are only filtered with a high-pass filter out of necessity to be able to integrate them, while the integration process in itself also works as a low-pass filter. As described in Section 3.5.5, there have been proposed a wide range of alternative filtering techniques for these kinds of data. For example, *wavelet decomposition* (Keenan and Wilhelm 2005), *adaptive filtering*

(Keenan and Wilhelm 2005), *noise discrimination* (Retory et al. 2016), and *empirical mode decomposition* (Liu et al. 2013). All of these techniques have been proven to be very effective at filtering motion artifacts from the signals. However, since motion artifacts are usually not a big concern related to sleep signals, it is not clear how much these filtering techniques improve the quality of these signals. Nevertheless, the effect of different kinds of filtering techniques is definitively an important aspect to study further.

We barely touched upon the effect of sensor misplacement during the work in this thesis (Section 5.2.2). The placement of the sensors does indeed affect the amplitude of the breaths, where the degree depends upon the distance from the optimal placement. For the signals captured from the external subjects, we positioned the sensors in their optimal locations. As the long-term goal is to allow people to use the sensors by themselves in the comfort of their own home, misplacements are very likely to occur. Consequently, the effect of misplacement on the signal quality is very important to study further.

Bibliography

Adams, Jose A., Ignacio A. Zabaleta, David Stroh, and Marvin A. Sackner. 1993. "Measurement of breath amplitudes: Comparison of three noninvasive respiratory monitors to integrated pneumotachograph." *Pediatric Pulmonology* 16, no. 4 (October): 254–258. ISSN: 87556863. doi:10.1002/ppul.1950160408.

Ahmadi, Negar, Sharon A. Chung, Alison Gibbs, and Colin M. Shapiro. 2008. "The Berlin questionnaire for sleep apnea in a sleep clinic population: Relationship to polysomnographic measurement of respiratory disturbance." *Sleep and Breathing* 12, no. 1 (February): 39–45. ISSN: 15209512. doi:10.1007/s11325-007-0125-y.

Alaska Sleep Clinic. 2015. "The 3 Types of Sleep Apnea Explained: Obstructive, Central, & Mixed." Accessed February 14, 2018. http://www.alaskasleep.com/blog/types-of-sleep-apnea-explained-obstructive-central-mixed.

Almazaydeh, Laiali, Khaled Elleithy, and Miad Faezipour. 2012. "Detection of obstructive sleep apnea through ECG signal features." In *IEEE International Conference on Electro Information Technology,* 1–6. IEEE, May. ISBN: 9781467308199. doi:10.1109/EIT.2012.6220730.

American Sleep Apnea Association. 2018a. "Central Sleep Apnea." Accessed February 13, 2018. https://www.sleepapnea.org/learn/sleep-apnea/central-sleep-apnea/.

———. 2018b. "Obstructive Sleep Apnea." Accessed February 14, 2018. https://www.sleepapnea.org/learn/sleep-apnea/obstructive-sleep-apnea/.

Askham, N, D Cook, M Doyle, H Fereday, M Gibson, U Landbeck, R Lee, C Maynard, G Palmer, and J Schwarzenbach. 2013. "The Six Primary Dimensions for Data Quality Assessment." *Group, DAMA UK Working:* 16.

Atkielski, Anthony. 2007. "Schematic diagram of normal sinus rhythm for a human heart as seen on ECG." Accessed February 15, 2018. https://commons.wikimedia.org/wiki/File:SinusRhythmLabels.svg.

Baker, Clark R Jr, and Edward M Richards. 2005. "Signal quality metrics design for qualifying data for a physiological monitor." US Patent 7,006,856.

Berry, Richard B., Rohit Budhiraja, Daniel J. Gottlieb, David Gozal, Conrad Iber, Vishesh K. Kapur, Carole L. Marcus, et al. 2012. "Rules for scoring respiratory events in sleep: Update of the 2007 AASM manual for the scoring of sleep and associated events." *Journal of Clinical Sleep Medicine* 8 (5): 597–619. ISSN: 15509389. doi:10.5664/jcsm.2172.

Berry, Richard B., and Mary H. Wagner. 2014. *Sleep Medicine Pearls,* 1–690. ISBN: 9781455770519.

biosignalsplux. 2018a. "biosignalsplux." Accessed March 1, 2018. http://biosignalsplux.com/en/.

———. 2018b. "RespiBAN Researcher." Accessed March 1, 2018. http://biosignalsplux.com/en/respiban-researcher.

BITalino. 2018a. "BITalino." Accessed January 27, 2018. http://bitalino.com.

———. 2018b. "BITalino PZT." Accessed January 27, 2018. https://store.plux.info/bitalino-sensors/40-respiration-pzt-sensor.html.

———. 2018c. "BITalino RIP." Accessed January 27, 2018. https://store.plux.info/professional-sensors/317-respiration-rip-820202501.html.

———. 2018d. "BITalino vs. biosignalsplux - intended use." Accessed March 1, 2018. http://bitalino.com/index.php/en/intended-use.

———. 2018e. "Plugged kit BLE." Accessed March 1, 2018. http://bitalino.com/en/plugged-kit-ble.

Boigelot, Denis. 2011. "Pearson correlation coefficient." Accessed February 7, 2018. https://upload.wikimedia.org/wikipedia/commons/d/d4/Correlation_examples2.svg.

Brainworks. 2018. "WHAT ARE BRAINWAVES?" Accessed February 15, 2018. http://www.brainworksneurotherapy.com/what-are-brainwaves.

Brouillette, Robert T, Anna S Morrow, Debra E Weese-Mayer, and Carl E Hunt. 1987. "Comparison of respiratory inductive plethysmography and thoracic impedance for apnea monitoring." *The Journal of Pediatrics* 111 (3): 377–383. ISSN: 00223476. doi:10.1016/S0022-3476(87)80457-2.

Cannesson, M., O. Desebbe, P. Rosamel, B. Delannoy, J. Robin, O. Bastien, and J. J. Lehot. 2008. "Pleth variability index to monitor the respiratory variations in the pulse oximeter plethysmographic waveform amplitude and predict fluid responsiveness in the operating theatre." *British Journal of Anaesthesia* 101, no. 2 (August): 200–206. ISSN: 00070912. doi:10.1093/bja/aen133.

Cantineau, J. P., P. Escourrou, R. Sartene, C. Gaultier, and M. Goldman. 1992. "Accuracy of respiratory inductive plethysmography during wakefulness and sleep in patients with obstructive sleep apnea." *Chest* 102 (4): 1145–1151. ISSN: 00123692. doi:10.1378/chest.102.4.1145.

CleveMed. 2018. "Type I, Type II, Type III Sleep Monitors, CMS AASM Guidelines." Accessed February 18, 2018. https://clevemed.com/cms-aasm-guidelines-for-sleep-monitors-type-i-type-ii-type-iii/.

Cohn, M A, A. S. V. Rao, M Broudy, S Birch, H Watson, Neal Atkins, Brian Davis, F D Stott, and Marvin A Sackner. 1982. "The respiratory inductive plethysmograph: a new non-invasive monitor of respiration." *Bulletin européen de physiopathologie respiratoire* 18 (4): 643–58. ISSN: 0395-3890.

Datatilsynet. 2015. “Anonymisering av personopplysninger. Veileder.”

Estrada, E, H Nazeran, J Barragan, J R Burk, E A Lucas, and K Behbehani. 2006. “EOG and EMG: Two important switches in automatic sleep stage classification.” In *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings,* 1:2458–2461. IEEE, August. ISBN: 1424400325. doi:10.1109/IEMBS.2006.260075.

Garcia-Molina, Hector, Jeffrey D Ullman, Jennifer Widom, Prentice Hall, and Stanford W Ascherman. 2008. *Database Systems: The Complete Book.* ISBN: 0136067018.

George, C. F., T. W. Millar, and M. H. Kryger. 1988. “Sleep apnea and body position during sleep.” *Sleep* 11, no. October 1987 (January): 90–99. ISSN: 0161-8105. doi:10.1093/sleep/11.1.90.

Gjøby, Svein Petter. 2016. “Extensible data acquisition tool for Android.” Master’s Thesis.

Gupta, Amit K. 2011. “Respiration rate measurement based on impedance pneumography.” *Texas Instruments application report SBAA181.*

Harvard. 2011. Accessed February 15, 2018. `http://healthysleep.med.harvard.edu/sleep-apnea/diagnosing-osa/understanding-results`.

Houston Sleep. 2018. “EMG (Electromyography) Testing:” accessed February 15, 2018. `http://www.houstonsleep.net/HTML/EMG.htm`.

How Equipment Works. 2018. “How pulse oximeters work explained simply.” Accessed February 15, 2018. `https://www.howequipmentworks.com/pulse_oximeter`.

Hrubos-Strøm, Harald, Anna Randby, Silje K. Namtvedt, Håvard A. Kristiansen, Gunnar Einvik, Juratešaltyte Benth, Virend K. Somers, et al. 2011. “A Norwegian population-based study on the risk and prevalence of obstructive sleep apnea The Akershus Sleep Apnea Project (ASAP).” *Journal of Sleep Research* 20 (1 PART II): 162–170. ISSN: 09621105. doi:10.1111/j.1365-2869.2010.00861.x.

Huang, Qi Rong, Zhenxing Qin, Shichao Zhang, and Chin Moi Chow. 2008. “Clinical patterns of obstructive sleep apnea and its comorbid conditions: a data mining approach.” *Journal of clinical sleep medicine : JCSM : official publication of the American Academy of Sleep Medicine* 4 (6): 543–50. ISSN: 1550-9389.

Johns, Murray W. 1991. “A new method for measuring daytime sleepiness: The Epworth sleepiness scale.” *Sleep* 14, no. 6 (November): 540–545. ISSN: 01618105. doi:10.1093/sleep/14.6.540.

Katz, Amiram, and Dudley S Dinner. 1992. “The effect of sleep position on the diagnosis of obstructive sleep apnea: a word of caution.” *Cleve.Clin.J Med.* 59 (6): 634–636.

Keenan, D. Barry, and Frank H. Wilhelm. 2005. “Adaptive and wavelet filtering methods for improving accuracy of respiratory measurement.” *Biomedical Sciences Instrumentation* 41 (0067-8856 (Print)): 37–42. ISSN: 00678856.

Kent State University. 2018. “SPSS Tutorials: Pearson Correlation.” Accessed February 15, 2018. `https://libguides.library.kent.edu/SPSS/PearsonCorr`.

Kogan, Dmitriy, A. Jain, S. Kimbro, Guillermo Gutierrez, and Vivek Jain. 2016. "Respiratory Inductance Plethysmography Improved Diagnostic Sensitivity and Specificity of Obstructive Sleep Apnea." *Respiratory Care* 61 (8): 1033–1037. ISSN: 0020-1324. doi:10.4187/respcare.04436.

Konno, K, and Jere Mead. 1967. "Measurement of the separate volume changes of rib cage and abdomen during breathing." *Journal of applied physiology (Bethesda, Md. : 1985)* 22 (3): 407–422. ISSN: 0021-8987.

Kristiansen, Stein, Mari Sønsteby Hugaas, Vera Goebel, Thomas Plagemann, Konstantinos Nikolaidis, and Knut Liestøl. 2018. "Data Mining for Patient Friendly Apnea Detection." *submitted to IEEE Access, May 2018.*

Liu, Shaopeng, Robert X Gao, Dinesh John, John Staudenmayer, and Patty Freedson. 2013. "Tissue artifact removal from respiratory signals based on empirical mode decomposition." *Annals of Biomedical Engineering* 41, no. 5 (May): 1003–1015. ISSN: 00906964. doi:10.1007/s10439-013-0742-5.

Lovdata. 2000. *Personopplysningsloven.* Accessed February 18, 2018. https://lovdata.no/dokument/NL/lov/2000-04-14-31.

M'henni, Habib. 2010. "Illustration of Obstructed Airways." Accessed February 14, 2018. https://commons.wikimedia.org/wiki/File%3AObstruction_ventilation_apn%C3%A9e_sommeil.svg.

matplotlib. 2017. "matplotlib." Accessed January 30, 2018. https://matplotlib.org.

MayoClinic. 2014. "EEG (electroencephalogram)." Accessed February 15, 2018. https://www.mayoclinic.org/tests-procedures/eeg/about/pac-20393875.

———. 2015. "Sleep apnea." Accessed February 13, 2018. https://www.mayoclinic.org/diseases-conditions/sleep-apnea/symptoms-causes/syc-20377631.

———. 2018a. "Continuous positive airway pressure (CPAP)." Accessed February 14, 2018. https://www.mayoclinic.org/diseases-conditions/sleep-apnea/multimedia/continuous-positive-airway-pressure-cpap/img-20007977.

———. 2018b. "Electrocardiogram (ECG or EKG)." Accessed February 15, 2018. https://www.mayoclinic.org/tests-procedures/ekg/about/pac-20384983.

Mazeika, G G. 2005. "GASP: A self-admistered screening questionnaire for Obstructive Sleep Apnea." In *SLEEP,* vol. 28, A325–A325.

McGill University. 2018. "Topics in Respiratory Physiology for Undergraduate Students." Accessed February 11, 2018. https://www.medicine.mcgill.ca/physio/resp-web/jfig5-1.htm.

McNicholas, Walter T. 2013. "New Standards and Guidelines for Drivers with Obstructive Sleep Apnoea syndrome," no. SEPTEMBER: 1–49. doi:10.13140/RG.2.1.4510.5129.

Minitab. 2017. "R-Squared: Sometimes, a Square is just a Square." Accessed February 16, 2018. http://blog.minitab.com/blog/statistics-and-quality-data-analysis/r-squared-sometimes-a-square-is-just-a-square.

Morgenthaler, Timothy I., Vadim Kagramanov, Viktor Hanak, and Paul A. Decker. 2006. *Complex sleep apnea syndrome: Is it a unique clinical syndrome?,* September. doi:10.1093/sleep/29.9.1203.

National Heart Lung and Blood Institute. 2013. "The Traditional Polysomnography." Accessed February 16, 2018. https://commons.wikimedia.org/wiki/File:Sleep_studies.jpg.

NOX Medical. 2018a. "NOX T3." Accessed January 27, 2018. http://www.noxmedical.com/products/nox-t3-sleep-monitor.

———. 2018b. "Noxturnal." Accessed January 27, 2018. http://www.noxmedical.com/products/noxturnal-software.

NumPy. 2017. "NumPy." Accessed January 30, 2018. http://www.numpy.org.

Oxford English Dictionary. 2018a. ""quality, n. and adj."."

———. 2018b. "plethysmography, n."

pandas. 2018. "pandas." Accessed January 30, 2018. https://pandas.pydata.org.

Parikh, Rajul, Annie Mathai, Shefali Parikh, G Chandra Sekhar, and Ravi Thomas. 2008. "Understanding and using sensitivity, specificity and predictive values." *Indian journal of ophthalmology* 56 (1): 45.

Pennock, Bernard E. 1990. "Rib cage and abdominal piezoelectric film belts to measure ventilatory airflow." *J Clin Monit* 6, no. 4 (October): 276–283. ISSN: 0748-1977. doi:10.1007/BF02842487.

PLUX. 2018. "PLUX." Accessed March 1, 2018. https://plux.info/index.php/en/.

Punjabi, N. M. 2008. "The Epidemiology of Adult Obstructive Sleep Apnea." *Proceedings of the American Thoracic Society* 5, no. 2 (February): 136–143. ISSN: 1546-3222. doi:10.1513/pats.200709-155MG.

ResMed. 2018. "ApneaLink Plus." Accessed February 17, 2018. https://www.resmed.com/us/en/healthcare-professional/products/diagnostics/apnealink-plus.html.

Retory, Yann, Pauline Niedzialkowski, Carole De Picciotto, Marcel Bonay, and Michel Petitjean. 2016. "New respiratory inductive plethysmography (RIP) method for evaluating ventilatory adaptation during mild physical activities." *PLoS ONE* 11 (3): e0151983. ISSN: 19326203. doi:10.1371/journal.pone.0151983.

scikit-learn. 2017. "scikit-learn." Accessed January 30, 2018. http://scikit-learn.org.

Scilingo, Enzo Pasquale, Antonio Lanatà, and Alessandro Tognetti. 2011. "Sensors for wearable systems." In *Wearable Monitoring Systems,* 3–25. Boston, MA: Springer US. ISBN: 9781441973832. doi:10.1007/978-1-4419-7384-9_1. arXiv: arXiv:1011.1669v3.

SciPy. 2018. "SciPy." Accessed January 30, 2018. https://www.scipy.org.

Seppänen, Tiina M, Olli-Pekka Alho, and Tapio Seppänen. 2013. "Reducing the airflow waveform distortions from breathing style and body position with improved calibration of respiratory effort belts." *Biomedical engineering online* 12, no. 1 (September): 97. ISSN: 1475-925X. doi:10.1186/1475-925X-12-97.

Shahid, Azmeh, Kate Wilkinson, Shai Marcu, and Colin M. Shapiro. 2011. "STOP-Bang Questionnaire." In *STOP, THAT and One Hundred Other Sleep Scales,* 371–383. New York, NY: Springer New York. ISBN: 9781441998934. doi:10.1007/978-1-4419-9893-4_92.

Shimmer. 2018a. "Shimmer." Accessed January 27, 2018. http://www.shimmersensing.com/about/.

———. 2018b. "Shimmer ECG." Accessed January 27, 2018. http://www.shimmersensing.com/products/ecg-development-kit.

Silva, Hugo Plácido da, Carlos Carreiras, André Lourenço, Ana Fred, Rui César das Neves, and Rui Ferreira. 2015. "Off-the-person electrocardiography: performance assessment and clinical correlation." *Health and Technology* 4, no. 4 (April): 309–318. ISSN: 21907196. doi:10.1007/s12553-015-0098-y.

Sleep Apnea Guide. 2018. "Central Sleep Apnea." Accessed February 14, 2018. http://www.sleep-apnea-guide.com/central-sleep-apnea.html.

Spearman, Charles. 1904. "The proof and measurement of association between two things." *The American journal of psychology* 15 (1): 72–101.

Stellwagen, Eric. 2011. "Forecasting 101: A Guide to Forecast Error Measurement Statistics and How to Use Them." Accessed February 21, 2018. http://www.forecastpro.com/Trends/forecasting101August2011.html.

Stevens, S.S. 1946. "On the Theory of Scales of Measurement." *Science* 103 (2684): 677–680. ISSN: 0036-8075. doi:10.1126/science.103.2684.677.

SweetZpot. 2018. "FLOW." Accessed April 25, 2018. https://www.sweetzpot.com/flow.

Talke, Pekka, Ray J. Nichols, and Daniel L. Traber. 1990. "Does measurement of systolic blood pressure with a pulse oximeter correlate with conventional methods?" *Journal of clinical monitoring* 6, no. 1 (January): 5–9. ISSN: 0748-1977. doi:10.1007/BF02832176.

The Mathworks, Inc. 2016a. *findpeaks - R2016b.* Natick, Massachusetts. Accessed January 26, 2018. https://se.mathworks.com/help/signal/ref/findpeaks.html.

———. 2016b. *MATLAB.* Natick, Massachusetts. Accessed February 1, 2018. https://se.mathworks.com/products/matlab.html.

———. 2018. *mathworks.* Natick, Massachusetts. Accessed February 1, 2018. https://se.mathworks.com/.

Tournade, Yoan. 2015. "Peak Detection in the Python World." Accessed January 30, 2018. https://blog.ytotech.com/2015/11/01/findpeaks-in-python/.

Tripathi, Manjari. 2008. “Technical notes for digital polysomnography recording in sleep medicine practice.” *Annals of Indian Academy of Neurology* 11, no. 2 (April): 129–138. ISSN: 1998-3549. doi:10.4103/0972-2327.41887.

Van Dongen, Hans P.A., Greg Maislin, Janet M. Mullington, and David F. Dinges. 2003. “The cumulative cost of additional wakefulness: Dose-response effects on neurobehavioral functions and sleep physiology from chronic sleep restriction and total sleep deprivation.” *Sleep* 26, no. 2 (March): 117–126. ISSN: 01618105. doi:10.1093/sleep/26.2.117.

Vaughn, Courtney M, and Pamela Clemmons. 2012. “Piezoelectric belts as a method for measuring chest and abdominal movement for obstructive sleep apnea diagnosis.” *Neurodiagnostic Journal* 52 (3): 275–280. ISSN: 23758627. doi:10.1080/21646821.2012.11079862.

Whyte, K F, M Gugger, G A Gould, J Molloy, P K Wraith, and N J Douglas. 1991. “Accuracy of respiratory inductive plethysmograph in measuring tidal volume during sleep.” *Journal of applied physiology (Bethesda, Md. : 1985)* 71, no. 5 (November): 1866–1871. ISSN: 01617567.

Wikipedia. 2018a. “Butterworth filter.” Accessed January 30, 2018. https://en.wikipedia.org/w/index.php?title=Butterworth_filter.

———. 2018b. “Mean percentage error.” Accessed April 14, 2018. https://en.wikipedia.org/wiki/Mean_percentage_error.

———. 2018c. “Strain gauge.” Accessed April 25, 2018. https://en.wikipedia.org/wiki/Strain_gauge.

Willmott, Cort J, and Kenji Matsuura. 2005. “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance.” *Climate research* 30 (1): 79–82.

Wu, Dan, Lei Wang, Yuan Ting Zhang, Bang Yu Huang, Bo Wang, Shao Jie Lin, and Xiao Wen Xu. 2009. “A wearable respiration monitoring system based on digital respiratory inductive plethysmography.” In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009,* 4844–4847. IEEE, September. ISBN: 9781424432967. doi:10.1109/IEMBS.2009.5332665.

Young, Terry, James Skatrud, and Paul E Peppard. 2004. “Risk Factors for Obstructive Sleep Apnea in Adults.” 291 (16): 2013–2016.

Appendices

Appendix A

Source Code

The source code presented and used in this thesis, as well as the automatic timekeeper application used during the signal capture procedure, can be found at: `https://github.uio.no/CESAR/Fredrik-L-berg`

Appendix B

Experiment Results

The raw data for the all the signal quality metrics are presented in the following tables. For the BITalino signals, the sensitivity is shown in Table B.1 and Table B.2, the PPV in Table B.3 and Table B.4, the CMP in Table B.5 and Table B.6, and the WAPE metric in Table B.7 and Table B.8. For Shimmer, all metrics are shown in Table B.9 and Table B.10. For RespiBAN, all metric scores are shown in Table B.11 and Table B.12, and all results for Flow are shown in Table B.13 and Table B.14.

Subjects	**Abdomen**	**Thorax**	**Sum**	**Abdomen$_{\text{int}}$**	**Thorax$_{\text{int}}$**	**Sum$_{\text{int}}$**
Subject 1	100.00%	99.40%	99.40%	97.01%	86.83%	85.03%
Subject 2	100.00%	97.87%	100.00%	98.94%	94.68%	95.79%
Subject 3	100.00%	100.00%	100.00%	100.00%	97.03%	100.00%
Subject 4	100.00%	100.00%	100.00%	99.12%	99.12%	100.00%
Subject 5	100.00%	97.18%	100.00%	98.60%	93.57%	99.29%
Subject 6	99.19%	95.16%	98.45%	96.77%	87.10%	94.49%
Subject 7	99.25%	100.00%	99.25%	94.03%	96.21%	98.48%
Subject 8	100.00%	97.98%	98.97%	100.00%	98.88%	98.98%
Subject 9	99.06%	95.33%	99.06%	73.39%	87.85%	94.29%
Subject 10	100.00%	99.37%	100.00%	98.08%	99.36%	100.00%
Subject 11	97.83%	100.00%	100.00%	93.48%	95.56%	98.90%
Subject 12	100.00%	87.33%	98.64%	100.00%	82.00%	92.47%

**int: integrated*

Table B.1: BITalino sensitivity results — supine position

Subjects	**Abdomen**	**Thorax**	**Sum**	**Abdomen$_{\text{int}}$**	**Thorax$_{\text{int}}$**	**Sum$_{\text{int}}$**
Subject 1	99.35%	89.17%	99.29%	98.03%	57.50%	98.58%
Subject 2	100.00%	100.00%	98.81%	97.59%	96.43%	97.59%
Subject 3	100.00%	100.00%	100.00%	98.72%	96.15%	98.72%
Subject 4	100.00%	-	100.00%	100.00%	-	92.41%
Subject 5	98.51%	97.76%	100.00%	95.52%	94.03%	98.50%
Subject 6	99.19%	98.25%	98.25%	98.26%	81.58%	93.86%
Subject 7	97.48%	100.00%	99.16%	92.44%	98.32%	94.96%
Subject 8	100.00%	100.00%	100.00%	98.77%	97.53%	100.00%
Subject 9	100.00%	98.88%	100.00%	96.70%	95.45%	97.75%
Subject 10	-	-	-	-	-	-
Subject 11	97.10%	98.53%	98.55%	96.97%	95.52%	97.10%
Subject 12	99.10%	95.50%	98.11%	83.64%	93.64%	95.33%

**int: integrated*

Table B.2: BITalino sensitivity results — side position

Subjects	**Abdomen**	**Thorax**	**Sum**	**Abdomen$_{\mathbf{int}}$**	**Thorax$_{\mathbf{int}}$**	**Sum$_{\mathbf{int}}$**
Subject 1	97.09%	99.40%	99.40%	99.39%	100.00%	99.30%
Subject 2	96.94%	95.83%	100.00%	100.00%	100.00%	100.00%
Subject 3	98.00%	97.09%	98.99%	97.03%	100.00%	98.99%
Subject 4	100.00%	96.58%	98.26%	98.26%	100.00%	100.00%
Subject 5	98.61%	93.88%	97.92%	100.00%	97.04%	99.29%
Subject 6	96.09%	86.13%	92.03%	100.00%	95.58%	93.75%
Subject 7	91.03%	95.00%	95.68%	93.33%	96.95%	98.48%
Subject 8	97.00%	97.98%	100.00%	100.00%	100.00%	100.00%
Subject 9	86.78%	89.47%	84.68%	74.07%	94.00%	94.29%
Subject 10	99.36%	97.52%	100.00%	100.00%	100.00%	100.00%
Subject 11	97.83%	91.84%	96.81%	98.85%	100.00%	98.90%
Subject 12	96.67%	95.62%	93.55%	100.00%	97.62%	98.54%

**int: integrated*

Table B.3: BITalino positive predictive value results — supine position

Subjects	**Abdomen**	**Thorax**	**Sum**	**Abdomen$_{\mathbf{int}}$**	**Thorax$_{\mathbf{int}}$**	**Sum$_{\mathbf{int}}$**
Subject 1	95.60%	90.68%	93.33%	98.68%	80.23%	97.89%
Subject 2	93.33%	97.70%	100.00%	93.10%	98.78%	100.00%
Subject 3	100.00%	98.73%	100.00%	100.00%	98.68%	100.00%
Subject 4	100.00%	-	94.05%	96.34%	-	97.33%
Subject 5	97.06%	90.34%	97.10%	100.00%	100.00%	100.00%
Subject 6	96.61%	80.00%	86.82%	96.58%	93.00%	92.24%
Subject 7	95.08%	98.35%	97.52%	97.35%	97.50%	98.26%
Subject 8	94.19%	97.59%	100.00%	100.00%	98.75%	100.00%
Subject 9	91.09%	87.13%	89.90%	92.63%	97.67%	95.60%
Subject 10	-	-	-	-	-	-
Subject 11	85.90%	95.71%	98.55%	94.12%	96.97%	100.00%
Subject 12	83.33%	86.18%	84.55%	93.88%	94.50%	90.27%

**int: integrated*

Table B.4: BITalino positive predictive value results — side position

Subjects	**Abdomen**	**Thorax**	**Sum**	**Abdomen$_{\text{int}}$**	**Thorax$_{\text{int}}$**	**Sum$_{\text{int}}$**
Subject 1	44.44%	77.78%	77.78%	66.67%	44.44%	44.44%
Subject 2	66.67%	55.56%	100.00%	88.89%	66.67%	66.67%
Subject 3	77.78%	66.67%	88.89%	66.67%	66.67%	88.89%
Subject 4	100.00%	55.56%	88.89%	66.67%	88.89%	100.00%
Subject 5	71.43%	14.29%	71.43%	85.71%	28.57%	85.71%
Subject 6	37.50%	0.00%	22.22%	62.50%	0.00%	33.33%
Subject 7	22.22%	33.33%	44.44%	33.33%	44.44%	55.56%
Subject 8	77.78%	55.56%	88.89%	100.00%	87.50%	89.89%
Subject 9	22.22%	22.22%	22.22%	0.00%	0.00%	33.33%
Subject 10	88.89%	44.44%	100.00%	77.78%	88.89%	100.00%
Subject 11	66.67%	33.33%	77.78%	55.56%	77.78%	77.78%
Subject 12	55.56%	33.33%	33.33%	100.00%	22.22%	22.22%

**int: integrated*

Table B.5: BITalino clean minute proportion results — supine position

Subjects	**Abdomen**	**Thorax**	**Sum**	**Abdomen$_{\text{int}}$**	**Thorax$_{\text{int}}$**	**Sum$_{\text{int}}$**
Subject 1	28.57%	0.00%	57.14%	57.14%	0.00%	57.14%
Subject 2	57.14%	85.71%	85.71%	28.57%	71.43%	71.43%
Subject 3	100.00%	85.71%	100.00%	85.71%	57.14%	85.71%
Subject 4	100.00%	-	71.43%	71.43%	-	28.57%
Subject 5	57.14%	14.29%	85.71%	42.86%	57.14%	71.43%
Subject 6	57.14%	42.86%	14.29%	42.86%	14.29%	28.57%
Subject 7	42.86%	85.71%	71.43%	42.86%	57.14%	42.86%
Subject 8	71.43%	71.43%	100.00%	85.71%	71.43%	100.00%
Subject 9	42.86%	14.29%	42.86%	28.57%	42.86%	57.14%
Subject 10	-	-	-	-	-	-
Subject 11	14.29%	57.14%	71.43%	42.86%	42.86%	85.71%
Subject 12	14.29%	14.29%	14.29%	28.57%	14.29%	0.00%

**int: integrated*

Table B.6: BITalino clean minute proportion results — side position

Subjects	**Abdomen**	**Thorax**	**Sum**	**Abdomen$_{\text{int}}$**	**Thorax$_{\text{int}}$**	**Sum$_{\text{int}}$**
Subject 1	26.24%	17.25%	25.13%	39.54%	16.05%	28.51%
Subject 2	10.78%	23.42%	10.56%	13.99%	40.20%	16.98%
Subject 3	10.62%	22.30%	10.00%	16.11%	38.72%	16.84%
Subject 4	8.38%	16.46%	7.35%	9.31%	24.86%	9.00%
Subject 5	7.51%	16.13%	10.26%	13.18%	24.87%	14.29%
Subject 6	13.29%	31.54%	17.19%	14.31%	44.35%	26.17%
Subject 7	20.95%	18.78%	13.56%	25.98%	23.75%	16.59%
Subject 8	11.51%	13.96%	12.44%	16.33%	24.61%	18.12%
Subject 9	24.41%	23.28%	24.65%	38.71%	25.24%	28.91%
Subject 10	9.68%	13.83%	9.05%	13.92%	14.78%	10.37%
Subject 11	16.93%	27.64%	15.06%	21.12%	30.38%	22.10%
Subject 12	5.65%	22.71%	16.11%	9.86%	33.27%	23.42%

**int: integrated*

Table B.7: BITalino breath amplitude accuracy (WAPE) results — supine position

Subjects	**Abdomen**	**Thorax**	**Sum**	**Abdomen$_{\text{int}}$**	**Thorax$_{\text{int}}$**	**Sum$_{\text{int}}$**
Subject 1	17.39%	30.06%	17.75%	28.61%	61.39%	27.81%
Subject 2	9.68%	17.41%	18.72%	22.44%	26.23%	23.35%
Subject 3	11.65%	26.29%	13.27%	18.23%	45.73%	18.20%
Subject 4	12.91%	-	13.46%	17.49%	-	23.30%
Subject 5	17.70%	24.57%	10.54%	17.66%	36.82%	14.32%
Subject 6	17.68%	31.54%	25.71%	21.32%	39.79%	28.95%
Subject 7	21.89%	8.70%	17.52%	21.89%	21.32%	22.88%
Subject 8	8.70%	18.83%	8.74%	16.30%	41.29%	12.55%
Subject 9	15.46%	24.94%	17.03%	21.37%	32.68%	24.41%
Subject 10	-	-	-	-	-	-
Subject 11	23.38%	22.59%	16.16%	28.09%	28.69%	23.94%
Subject 12	25.26%	18.74%	22.78%	27.00%	29.23%	28.90%

**int: integrated*

Table B.8: BITalino breath amplitude accuracy (WAPE) results — side position

Subjects	**Sensitivity**	**PPV**	**CMP**	**WAPE**
Subject 1	100.00%	100.00%	100.00%	13.20%
Subject 2	100.00%	98.95%	88.89%	25.94%
Subject 3	100.00%	100.00%	100.00%	20.54%
Subject 4	100.00%	95.76%	77.78%	6.93%
Subject 5	100.00%	100.00%	100.00%	12.55%
Subject 6	97.67%	94.03%	55.56%	23.08%
Subject 7	100.00%	100.00%	100.00%	8.92%
Subject 8	-	-	-	-
Subject 9	93.20%	87.27%	22.22%	26.54%
Subject 10	94.27%	95.48%	22.22%	17.87%
Subject 11	100.00%	92.23%	55.56%	16.64%
Subject 12	98.66%	98.66%	66.67%	13.58%

Table B.9: All Shimmer results — supine position

Subjects	**Sensitivity**	**PPV**	**CMP**	**WAPE**
Subject 1	95.30%	88.75%	0.00%	36.98%
Subject 2	85.71%	98.51%	66.67%	29.71%
Subject 3	100.00%	98.72%	85.71%	31.05%
Subject 4	97.50%	98.73%	57.14%	13.31%
Subject 5	100.00%	100.00%	100.00%	12.14%
Subject 6	100.00%	100.00%	100.00%	17.17%
Subject 7	100.00%	100.00%	100.00%	8.74%
Subject 8	100.00%	96.00%	71.43%	31.65%
Subject 9	-	-	-	-
Subject 10	-	-	-	-
Subject 11	97.22%	97.22%	50.00%	11.55%
Subject 12	-	-	-	-

Table B.10: All Shimmer results — side position

Subjects	**Sensitivity**	**PPV**	**CMP**	**WAPE**
Subject 1	96.35%	95.85%	44.44%	14.20%
Subject 2	98.68%	85.23%	44.44%	10.99%
Subject 3	99.02%	88.60%	66.67%	11.68%
Subject 4	97.87%	94.36%	55.56%	15.76%
Subject 5	100.00%	95.00%	77.78%	9.31%
Subject 6	96.38%	88.09%	11.11%	18.01%
Subject 7	99.18%	91.67%	55.56%	13.31%
Subject 8	98.21%	83.33%	11.11%	13.40%
Subject 9	97.78%	88.00%	55.56%	18.14%
Subject 10	100.00%	95.50%	66.67%	12.76%
Subject 11	98.99%	93.33%	55.56%	11.99%

Table B.11: All RespiBAN results — supine position

Subjects	**Sensitivity**	**PPV**	**CMP**	**WAPE**
Subject 1	95.95%	97.26%	42.86%	11.20%
Subject 2	100.00%	72.16%	14.29%	30.05%
Subject 3	100.00%	84.54%	57.14%	11.61%
Subject 4	96.45%	91.28%	42.86%	14.06%
Subject 5	100.00%	89.68%	57.14%	10.54%
Subject 6	99.07%	92.17%	28.57%	16.55%
Subject 7	98.84%	90.43%	42.86%	13.89%
Subject 8	100.00%	83.81%	42.86%	12.78%
Subject 9	98.67%	77.08%	28.57%	16.93%
Subject 10	98.72%	90.59%	57.14%	12.36%
Subject 11	100.00%	84.09%	71.43%	11.17%

Table B.12: All RespiBAN results — side position

Subjects	**Sensitivity**	**PPV**	**CMP**	**WAPE**
Subject 1	98.92%	98.92%	75.00%	11.27%
Subject 2	100.00%	100.00%	100.00%	10.60%
Subject 3	100.00%	100.00%	100.00%	5.92%
Subject 4	98.87%	98.87%	50.00%	9.13%
Subject 5	99.19%	97.60%	62.50%	11.05%
Subject 6	99.19%	97.60%	62.50%	10.65%
Subject 7	95.90%	100.00%	60.00%	7.54%
Subject 8	96.97%	98.97%	50.00%	12.89%
Subject 9	100.00%	99.06%	88.89%	5.19%
Subject 10	100.00%	99.05%	80.00%	5.04%
Subject 11	98.94%	96.88%	75.00%	6.96%

Table B.13: All Flow results — supine position

Subjects	**Sensitivity**	**PPV**	**CMP**	**WAPE**
Subject 1	97.87%	100.00%	83.33%	11.61%
Subject 2	91.43%	100.00%	50.00%	12.91%
Subject 3	100.00%	98.78%	83.33%	6.29%
Subject 4	100.00%	97.01%	50.00%	13.64%
Subject 5	98.99%	100.00%	83.33%	5.53%
Subject 6	95.74%	100.00%	83.33%	15.59%
Subject 7	96.43%	98.78%	25.00%	9.20%
Subject 8	100.00%	100.00%	100.00%	8.85%
Subject 9	100.00%	98.84%	85.71%	6.61%
Subject 10	100.00%	100.00%	100.00%	9.14%
Subject 11	100.00%	97.30%	71.43%	6.32%

Table B.14: All Flow results — side position