Bjørgan, Mads T.
Ulvøen, Thomas

Classifying Hypopnea and Obstructive Sleep Apnea with the Somnofy Doppler Radar

Master's thesis in Computer Science
Supervisor: Langseth, Helge
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of Science and Technology

vital things

Bjørgan, Mads T.
Ulvøen, Thomas

Classifying Hypopnea and Obstructive Sleep Apnea with the Somnofy Doppler Radar

Master's thesis in Computer Science
Supervisor: Langseth, Helge
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

The thesis explores whether Somnofy, a Doppler-based radar sensor, is applicable as a screening tool for subjects with possible sleep-disordered breathing . Sleep-disordered breathing is clinically diagnosed by a polysomnography study, which is costly both with regards to resources and time-consumption. It is therefore advantageous to investigate whether a highly scalable and portable radar sensor, such as Somnofy, can be used as a screening tool in order to reduce the usage of unnecessary PSG studies. A convolutional neural network is proposed, which provides predictions on a one-second granularity using a sliding window approach with a 30-second window. Various methods for treating data imbalance is investigated, and a feature importance study is conducted.

The proposed model is not precise enough to tell exactly when an event occurs but shows promising results for predicting the patients apnea-hypopnea index. A suggested hybrid sampling approach shows the most promising empirical results, while a feature importance study shows that oxygen saturation, respiration data, and physical movement are key features. Clinical application is not yet feasible, but the model shows promising and accurate results in assessing the severity of a possible diagnosis. Further validation and empirical testing are required before the proposed model is applicable for screening.

Preface

This master thesis was written during the spring semester of 2019 for the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU), in cooperation with VitalThings.

The subject of this paper was defined by Ståle Toften, acting representative for VitalThings, in cooperation with supervisor and professor Helge Langseth. We would like to express our gratitude to VitalThings and Toften for valuable feedback throughout the thesis, to which we owe a great deal of success. We would also like to extend our gratitude to supervisor Helge Langseth for his continuous initiative and assistance throughout the semester.

Mads Bjørgan, Thomas Ulvøen

Trondheim, June 6, 2019

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

High quality sleep is paramount in ensuring good quality of life and general good health. It is the process in which the human body and mind regenerate, and without it, no one is yet to endure. Despite this, the average person is surprisingly unaware of how much our sleep patterns affect our lives. Frequent sleep insufficiency has been shown to be strongly connected to physical distress, frequent mental distress, anxiety and poor general health (Strine and Chapman, 2005). Looking further into the reality of sleep-related diseases, specifically regarding sleep apnea, it is evident that there is an enormous cost to the society related to the effects and results of sleep disease (Hillman et al., 2006). Not only considering the direct health care costs like treatment and diagnosis, but productivity loss, associated conditions like depression and diabetes, work-related injuries and so on. Sleep is connected to most, if not all, aspects of human health.

Given that the different variations of sleep apnea are a highly prevalent disease in the general population (Young et al., 2002), the number of patients or subjects with a disease far outweighs the facilities and doctors capable of diagnosing and treating such diseases. The current standard for diagnosing sleep-related diseases is to conduct a sleep study in a sleep lab, where the patient is subject to a polysomnography test (PSG), that measures among other things heart and brain activity, breathing patterns, the saturation level of oxygen and body movement.

These tests are known to be expensive in both time and cost, and the patient is taken out of his normal sleep environment, which often results in a considerably larger variation between nights, a crucial factor for assessing sleep disease (Levendowski et al., 2009).

Several systems for in-home diagnostics of the apnea-related disease has in the past been successfully used, but almost all have in common that they require some sort of physical connection to the patient. Common connections are blood oximetry sensors on a finger, devices for measuring nasal airflow, or belts measuring chest or abdominal movement. All of the sensors above introduce some level of burden to the patient, and may not provide all the information needed to precisely assess the presence of apnea events. Using radar technology we aim to mitigate this burden, and to provide what will become a cost-effective and reliable home solution for detecting sleep apnea. In addition, we will provide to our knowledge the first radar-based system that is able to detect sleep apnea related events, pushing the frontier of how machine learning is used for medical research. If successful, the Somnofy solution could revolutionise the at-home medical industry, providing a significant contribution to society both in terms of health and socioeconomic benefit.

1.2 Goals and Research Questions

The goal of the research conducted in this thesis is given below. It is put in place to provide a general direction in order to guide the project throughout the semester.

Goal Classifying different variations of sleep-disordered breathing using data obtained from Somnofy, while achieving comparable results to automated PSG based systems.

The Doppler-based radar Somnofy, which monitors vital signs of a subject, is used to investigate the goal. Radar-based systems are not as precise as a Polysomnography in their ability to measure vital signs, but scales better and is suitable as part of a home-based screening program used to evaluate whether a Polysomnography should be conducted. In order to successfully achieve the previously stated goal, some research questions are put down in order to aid the process of designing and implementing the system.

Research questions 1 Which features are the most informative in the detection of the different variations of sleep-disordered breathing?

Research question 2 What is the de facto standard method to minimise the effects of highly imbalanced datasets in multiclass classification?

Research question 3 What are the current best performing approaches for classifying sleep-disordered breathing?

Research question 1 is motivated by the vast option of different features available to radar-based systems, and that radar-based features might be of a different value than comparable Polysomnography features. Research question 2 is motivated by the high degree of imbalance between non-event periods and hypopnea/apnea-events. Research question 3 is motivated by identifying the most effective approaches in the classification of sleep-disordered breathing, and how the respective research is applicable to radar-based systems.

1.3 Research Method

To reach the designated goal, and to answer the research questions set out, a literature study sleep-disordered breathing and time-series analysis are conducted, whereupon a model is suggested and empirically tested. A separate search is undertaken for determining which strategy is the most desirable when working with imbalanced datasets, and later empirically tested. An experiment involving random forests is conducted for investigating feature importance.

1.4 Thesis Structure

The remainder of the thesis has the following structure:

Chapter 2 Presents essential background theory within sleep, related diseases and the technology used in sleep monitoring.

Chapter 3 Provides an overview of relevant theory related to machine learning

Chapter 4 Provides an overview of the relevant state of the art theory for sleep-disordered breathing recognition.

Chapter 5 Presents the datasets used and its structures, in order to provide background for the choices made in Chapter 5 and 6.

Chapter 6 Presents the architecture and model used for recognising sleep-disordered breathing with the Somnofy radar.

Chapter 7 Presents the experiments and the results obtained with the model.

Chapter 8 Evaluation of the experiments, followed by an open discussion, before presenting the conclusion of the thesis.

Chapter 2

Background Theory and Motivation

This chapter provides the reader with required elementary knowledge on sleep, sleep-disordered breathing, and the technology used to diagnose sleep-related events such as apnea or hypopnea. This introduction is advisable to read for anyone unfamiliar on the topic, in order to follow the forthcoming chapters of the thesis. In addition, the technology behind Somnofy is presented to provide an understanding of the benefits and drawbacks of the solution.

Section 2.1 presents theory related to sleep itself. Sleep disorders, such as hypopnea and apnea, is presented in Section 2.2. Methods for monitoring vital signs, as Polysomnography and Somnofy, is presented in Section 2.3, while Section 2.4 presents the motivation for addressing this problem.

2.1 Sleep

The average person spends roughly one-third of their life sleeping. Although the focus of public media in the latest years is mostly directed towards the effect of sleep on your mind and mental status, it also affects most of the physiological systems in your body. The dynamic and complex process of sleep is a widely studied field, but much is still to be learned. In this chapter, a few key aspects of sleep

and sleep-related disorders are investigated, in order to establish a foundation for the research conducted in this thesis.

2.1.1 Sleep Stages

Sleep stages are the formal definition of the different stages of sleep, and the differences that separate them. The first official standardisation of sleep stages was put in place in 1968 (Rechtschaffen and Kales, 1968). The different stages were based on the changes seen in electroencephalography (EEG). EEG monitors brain activity by reading electronic signals, and measures brain waves, using electrodes connected to the scalp as visualised in Figure 2.1.

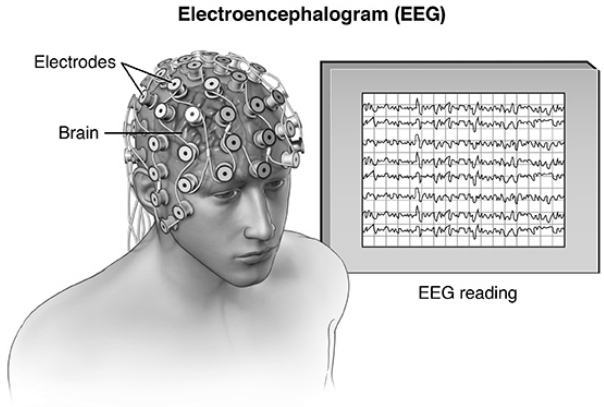


Figure 2.1: Visualisation of EEG. The EEG produces several distinct signals with information about the subjects brain activity.

EEG is a non-invasive procedure but requires both expensive equipment and time. In addition, the subject must be in a facility where the equipment is available. This is discussed further later on. The definition of sleep stages was based on the broad division between REM sleep and non-REM sleep, typically referred to as NREM. Further on, the original definition by Rechtschaffen and Kales (1968) divided NREM sleep into four stages: Stage I, II, III and IV.

In 2007, AASM (American Academy of Sleep Medicine) published a manual containing a revision of the guidelines for sleep classification (Conrad et al., 2007). The revision, which became the de facto definition for sleep stage classification,

attempted to simplify and clarify the scoring rules for both sleep stages and sleep associated diseases. One of the changes made was combining stage III and IV into stage N3. Hence, the new definition of sleep stages consisted of 4 stages: REM, N1, N2, N3. In addition, the state of being awake is often added in practical applications in order to capture the information of non-sleep during a night.

The characterisation of each sleep stage is given in Figure 2.1 to simplify the understanding of the difference between each stage. The values displayed are approximate values for a healthy subject, but note that the distribution of sleep stages is highly subjective.

Sleep stage	**Percentage**	**Description**
N1	3-5%	The stage between being awake and falling asleep. Heartbeat, breathing and eye movement slow down, and one might experience body twitches. Commonly known as light sleep.
N2	50-60%	The transition between light and deep sleep. Brain wave activity stops, but with occasional bursts of activity. Drop in body temperature and eye movement.
N3	10-20%	N3 is commonly known as deep sleep. Heartbeat and breathing is at its lowest. Cell regeneration, energy restoration and growth are important physiological processes during this sleep stage.
REM	10-25%	Know as REM due to the rapid eye movement observed when being in this state. Breathing is faster and often irregular, in addition to increased blood pressure. Dreaming most often occur in the REM stage.

Table 2.1: Overview of the different sleep stages (Conrad et al., 2007).

2.2 Sleep Disorders

Apnea is generally defined as a cessation of respiration for more than 10 seconds. It is seen as a component of several other defined apnea types, such as Central Sleep Apnea and Obstructive Sleep Apnea. However, note that apnea is a general

term, which can occur in different variations of sleep-related diseases for different reasons.

Hypopnea is another common term that is a part of sleep-disordered breathing. It is the reduction of airflow during sleep and is defined in this thesis by a ≥ 30 % reduction of nasal pressure from baseline, with an associated ≥ 3 % desaturation from pre-event baseline, using the 2008 revision by AASM (Ruehland et al., 2009). In addition, the event is required to last for ≥ 10 seconds.

The reader should be aware that the definition of hypopnea underwent a significant change in 2008. The requirements were previously ≥ 50 % airflow reduction, or an ≥ 3% oxygen desaturation. For future reference, it is often unclear in research related to sleep-disordered breathing which definition that is actually used.

In order to provide a general measure of the severity of a patient's apnea-related condition, the Apnea-Hypopnea Index (AHI) was put in place. The AHI is based on the number of events per hour of sleep and classifies a patient on four different levels of severity. The index is summarised in Table 2.2.

Type	Number of events
Normal	$\text{AHI} < 5$
Mild	$5 \leq \text{AHI} < 15$
Moderate	$15 \leq \text{AHI} < 30$
Severe	$30 \leq \text{AHI}$

Table 2.2: Severity categorisation of the Apnea-Hypopnea Index (AHI) (Ruehland et al., 2009)

OSA - Obstructive Sleep Apnea

Obstructive Sleep Apnea is a form of apnea where the primary requirement for diagnosis is 5 or more predominantly obstructive respiratory events, where the respiratory events are either a full or partial reduction of airflow in the upper airway. OSA may, therefore, contain either hypopnea- or apnea-events, depending on the severity of the collapse of the airway. The throats behaviour during normal, partially (hypopnea) and fully blocked (apnea) respiration is visualised in Figure 2.2.

CSA - Central Sleep Apnea

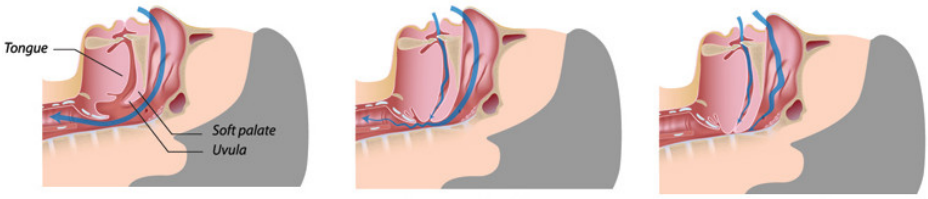


Figure 2.2: From left to right: Normal, partially blocked (hypopnea) and fully blocked (apnea) upper airway (Hudson Sleep & TMJ Center, 2019).

Central Sleep Apnea (Eckert et al., 2007) resembles OSA. While OSA resembles a hardware problem, CSA is more like a software problem. There are several different types of CSA, and in itself, CSA is a very generic term. In this thesis, there is made no distinction between them. In general, CSA is characterised by a lack of impulses from the brain telling the body to breathe during sleep. This can be further categorised into two broad types: Hypercapnic CSA and Non-Hypercapnic CSA. The most central difference is that hypercapnic CSA is associated with daytime hypoventilation. Daytime hypoventilation is by definition an increased concentration of carbon dioxide, which in other words most commonly means inadequate oxygen saturation levels.

An important difference during an OSA event is that, unlike CSA, the body still attempts to breathe, which is observable through the movement of the thorax (chest region) during an event.

MA - Mixed Apnea

Mixed, or Complex Apnea is a combination of one or more symptoms of OSA and CSA. There is no standard definition used in the literature on machine-aided detecting, and if a definition is given it is often quite vague. Here, an apnea event is defined as mixed if the following criteria are met:

- Event meets apnea criteria for a minimum of 2 breaths during
- Associated with absent respiratory effort during one portion of the event
- Presence of respiratory effort in the remaining portion

According to a study by the American Journal of Respiratory and Critical Care Medicine (Bixler et al., 2001), the prevalence of sleep-apnea was 3.9% for men and 1.2% for women. Those figures include patients with an AHI greater than

10 and reveal how many that are actually affected by the disease. Out of the variations above, OSA is the most common form of apnea, with CSA accounting for about 5-10% of clinic patients. Mixed apnea is closely related to OSA, and around 6.5% of diagnosed OSA patients show symptoms of Mixed or Complex apnea (Javaheri et al., 2009). It is clear that with such a large amount of the population being affected by these diseases, and such a high cost for diagnosis, it is of utmost relevance to invent solutions to screen for or diagnose sleep apnea at a potentially large scale.

2.3 Monitoring Vital Signs

Sleep-disordered breathing is not trivial to measure or detect directly but is predicted through a set of data of vital signs. Vital signs are merely signals read of the body, some well known such as heart rate (HR) and others more limited to medical professionals such as respiration rate (RR). The data in this thesis is collected from two sources, namely a Polysomnography (PSG) and the Somnofy radar. The devices and their signals are explored in detail in Section 2.3.1 and 2.3.2, for the PSG and Somnofy respectively.

2.3.1 Polysomnography

Polysomnography is the current gold standard for analysis and diagnosis in sleep studies. A PSG is performed overnight, and usually involves recording EEG (electroencephalogram), EOG (electrooculogram), ECG (electrocardiogram), airflow, thoracic and abdominal movements, and oximetry. Body movements, video and audio can be recorded as well.

A trained specialist mounts the equipment before the patient sleeps one or several nights to record data. The data is then scored by at least two specialists, and the resulting observations or analysis used by a physician to present a diagnosis.

While known as a gold standard for sleep analysis PSG is demanding in terms of both specialists and equipment, as well as being relatively intrusive for the patients compared to other alternatives such as HSAT (home sleep apnea tests). With that in mind, PSG analysis scales poorly when covering large populations and might not capture a realistic view of how the actually patient sleeps. However, few or none substitutes are available which can claim the same accuracy or

precision. The following segment describes the most common signals obtained in a PSG study.

Electroencephalogram EEG measures neurological activity in the brain, where a set of sensors are placed on the head. Figure 2.1 visualises an elementary concept of how it works.

Electrooculogram EOG is related to monitoring eye movement, where pairs of electrodes are placed around or on the side of the eye.

Electrocardiography ECG is the process of recording heart activity, where a set of electrodes is placed at central positions on the body. The electrodes read small electrical changes resulting from the heart pumping blood around the body, which yields an accurate reading of heart activity.

Air flow Air flow is typically measured by monitoring the nose and measures data related to breathing rates. The flow is measurable in various ways, but the most precise method is usually to mount a sensor on or around the nose.

Thoracic and abdominal movements Thoracic and abdominal events holds data related to the movement of the chest and upper abdomen wall, respectively, which is commonly used for reading respiratory data. This is recorded by wearing two separate belts, one for each measurement.

Oximetry Oximetry records oxygen saturation. A sensor is typically placed in front of the patient's nose or mouth while sleeping, or placed on the finger using absorption of light sent through the body.

2.3.2 Somnofy

Somnofy is a commercial sleep monitor under development by the start-up VitalThings. When placed on a nightstand or mounted on a nearby wall, Somnofy monitors vital signs and other signals such as movement, light, temperature, and sound, using radar technology and other sensors placed on the device. The radio chip itself is developed by Novelda, while VitalThings develops the monitoring and analysis algorithms on top of the data extracted from the radar. A prototype of the radar can be seen in Figure 2.3

The Somnofy product utilises the Novelda XeThru Ultrawideband Radar model X2 (Novelda, 2015). The X2 model is an IR-UWB pulse-Doppler radar, with

Figure 2.3: The Somnofy Radar device, here mounted on a table-stand.

a chip size of 5x5 mm making it highly suitable for small, lightweight devices. Pulse-Doppler radars working in the ultra-wideband (UWB) are able to penetrate softer materials like bed sheets, clothes, curtains, but also lighter or less dense forms of walls and structures (Stone, 1997). When hitting denser objects, the waves reflect back, allowing a detailed analysis of the collision object.

What allows for extracting movement from the data when using radar technology, is that a moving target will shift the frequency of the returned signal. This phenomenon is known as the Doppler effect. Due to the very precise and high-frequency sampling of the radar, even very small movements such as heartbeats are observable.

The radar itself performs various feature extraction from the raw sensor data. One of the most important features is the frequency of respiration, which is obtained by forming range-frequency-power matrices, referred to as pulse-Doppler matrices. To generate these matrices, a fast Fourier transform is performed on the data for each 5-cm increment of the total distance of the radar. To obtain respiration frequency, a relatively long Hanning window of 20 seconds is used, while as for body movement a Hanning window of either 3 or 20 seconds is used. Each second the fast Fourier transformation is repeated, which results in a measurement rate of 1 Hz.

The most relevant signals Somnofy can extract from subjects are currently body

movement, respiration rate, and heart rate. Following statements from VitalThings themselves, the body movement signal is rather sharp and precise. Respiration rate is also known for being rather precise with a small error margin of 2 percentage mean absolute error. Contrary, heart rate is still a signal that is under development, and currently not advisable for application.

A detailed description of which data that are obtained through Somnofy available for this thesis is found in Section 5.3. Let is also be noted that Somnofy has a Bluetooth integration, which in the future can be used for measuring oxygen saturation through a wireless Bluetooth finger cap.

2.4 Motivation

Sleep-related disorders are, as previously discussed, very common. In fact, a review of large epidemiological studies conducted in several different countries shows a median prevalence of obstructive sleep apnea of 22 %, defined with AHI $\geq$ 5 (Franklin and Lindberg, 2015). The numerous lifestyle-related diseases or medical events that are linked to sleep-related disorders, such as stroke, coronary artery disease, high blood pressure, and obesity, tells a tale of a severe medical condition. At the same time, sleep-related disorders such as apnea have been getting less attention than more commonly discussed diseases such as cardiovascular diseases.

There are several reasons why this is the case, one of them being that diagnosing sleep-related disorders are much more costly than for other common diseases. Spending one night at a sleep lab requires both spending time away from home, in addition to the cost of personnel and reserving sparse resources for sleep monitoring. Home tests exist, but are sparse and has limited trust within the medical community (Rosen et al., 2017). In addition, both sleep labs and current solutions for home sleep apnea test (HSAT), all require some form of attached apparatus for measuring, for example, nasal flow, blood saturation or electroencephalographic (EEG) signals. All these bring some sort of discomfort or unfamiliarity to the patients sleep environment and might be unsuited for patient groups such as young children.

In addition, there is a reason to believe substantial hidden figures exist with regards to sleep-related disorders, often because most people simply do not know they have or could have sleep apnea. As a statement of how common this problem

is, the subjects in both of the two trials providing data for this thesis were supposed to be healthy, but multiple subjects were diagnosed with sleep apnea as a result of the data collection.

A stronger method for diagnosing sleep apnea, comparable to that of sleep lab measurements, would lessen the considerable load on the medical community within sleep-related disorders. A non-contact based approach such as Somnofy, with its potential for home screening and initial diagnosis of sleep-related diseases, could with good results become a beneficial contribution to the health care industry.

In summary, sleep-disordered breathing is a fairly common disease, but where a majority remains undiagnosed. This is due to a multiple of reasons, but mainly because of the lack of convenient methods for diagnosing or screening patients for sleep-related diseases. The ones that exist, and acknowledge by industry, are time-consuming and requires an unscalable set of resources. This is a key motivation for exploring the possibilities of using Somnofy, a non-intrusive scalable radar monitor, for screening of sleep-disordered breathing.

Chapter 3

Machine Learning

Chapter 3 presents the required background theory related to machine learning and deep learning used in part of this thesis. The following material is assumed knowledge for the average student majoring in artificial intelligence but offers a basic introduction on the required theory for later discussion.

Section 3.1 presents the motivation for handling imbalanced datasets and a brief overview of the most trivial approaches. Accuracy is often a poor metric for evaluating models on imbalanced datasets. Section 3.2 discusses how to better quantify the performance of a model. A brief overview of convolutional neural networks are presented in Section 3.4, and recurrent neural networks in Section 3.5. Section 3.6 presents the theory behind Random Forests, while Section 3.7 presents theory related to k-Fold cross-validation, a robust resampling method used for model evaluation.

3.1 Imbalanced Datasets

Having an imbalanced dataset implies an uneven ratio between the occurrences of classes in the dataset. This is, unfortunately, a known problem in several domains, typically including computer vision, medical diagnosis, and fraud detection. Even a modest imbalance can have a significant detrimental effect on accuracy (Mazurowski et al., 2008), and hence, balancing data is a key challenge

in order to achieve a well-performing machine learning algorithm.

In the following subsections, the reader will find a set of traditional and trivial techniques used to compensate for imbalanced datasets, presented by Kotsiantis et al. (2006). Although the methods do not make up the complete picture of the available balancing techniques, it serves as an introduction to the most known approaches referring to literature in apnea and hypopnea detection, as seen later in 4.2.

In order to introduce the different sampling techniques, we introduce notation where $D_{original}$ is the original dataset, class K_{maj} the majority class, class K_{min} the minority classes and $D_{balanced}$ the balanced dataset.

Oversampling

Oversampling is a process where the size of the minority class(es) and the largest majority class is equivalated by enlarging the minority classes using copies of the existing data. With the introduced notation, the balanced dataset $D_{balanced}$ is generated by keeping the majority class K_{maj} and then randomly sampling from the minority class(es) K_{min} with replacement, until for all K_{min} in $D_{balanced}$ the size of K_{min} equals the size of K_{maj}. Although random sampling is not a particularly sophisticated or novel approach it is known for being effective, but might lead to overfitting (Chawla et al., 2002).

Undersampling

Instead of creating copies of the minority classes, undersampling reduces the size of the majority classes so that the size of all classes K_{maj} equal the size of the smallest minority class K_{min}. $D_{balanced}$ is created by randomly sampling the majority classes K_{maj} without replacement from $D_{original}$, until a given imbalance ratio is achieved. An imbalance ratio of 1 means all classes are sampled equally, while a ratio of 0.5 means K_{maj} is sampled twice for every K_{min} sample. Different variations are suggested in the literature, but a significant downside of undersampling is the loss of information when discarding data.

Cost-sensitive learning

Cost-sensitive learning (Domingos, 1999) is, contrary to both undersampling and oversampling, not adjusting the data corpus, but instead adjusting the cost function of the classifier. One example of an adjustment is to modify the weighting for misclassification of each class, often relative to the size of the class, so that classes with fewer cases are given a higher weighting.

Thresholding

Thresholding is another technique involving modification of the decision rule of the classifier. This approach is also known as *threshold moving* or *post scaling*, and is a process where output class probabilities are shifted. Indicated by the name post scaling, thresholding is only done during the test phase.

Adjustment of output class probabilities is possible in many ways. The most basic approach is to compensate for prior class probabilities before sampling (Richard and Lippmann, 1991), simply by counting the frequency of each class in the imbalanced dataset. The output is then shifted by dividing the output from the final layer by its estimated prior probability, as seen in equation 3.1.

The following equation presents a trivial approach for adjusting output class probabilities for thresholding

$$O_{t,c} = \frac{O_{o,c}}{P(c)}, \tag{3.1}$$

where $O_{t,c}$ is the output probability for class c when applying thresholding, and $O_{o,c}$ is the original output probability for class c, and $P(c)$ the prior probability for the occurrence of class c.

3.2 Measuring Performance on Unbalanced Predictions

There are several metrics that are useful when assessing performance on predictions of rare events or classes. To see why one needs to take extra measures when

assessing performance, imagine a situation where data from one night of relatively healthy sleep is given to a classifier. If 95% of the night can be classified as healthy, the classifier would quickly realise that when predicting every data point as healthy it achieves 95% accuracy. Despite this seemingly good performance, the model would be useless. To avoid this, several metrics exist to provide better information about the performance of the model. For the sake of completeness, all the metrics used in the process of evaluating the models are defined below.

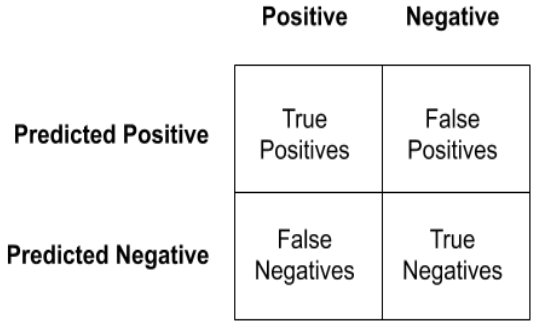


Figure 3.1: Confusion matrix for binary predictions, used when calculating metrics for performance on unbalanced class-distributions.

Accuracy calculates the overall accuracy of the model, in other words, what percentage of its predictions are correct.

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

Specificity measures how well the model is able to classify actual negative cases as such, and thus keeping false positives low.

$$Specificity = \frac{TP}{FP + TN}$$

Precision describes how precise the prediction of positive cases is, by calculating the ratio between correct positive predictions and all the positive predictions.

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

Recall measures how many of the positive cases the model is able to classify as positive.

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

Precision and recall do not individually provide much information about the performance of the model. As an example, getting a high recall-score would be easy to achieve if all cases were classified as positive. Combining the two measures, using a weighted average of precision and recall yields the F1 score or F-measure.

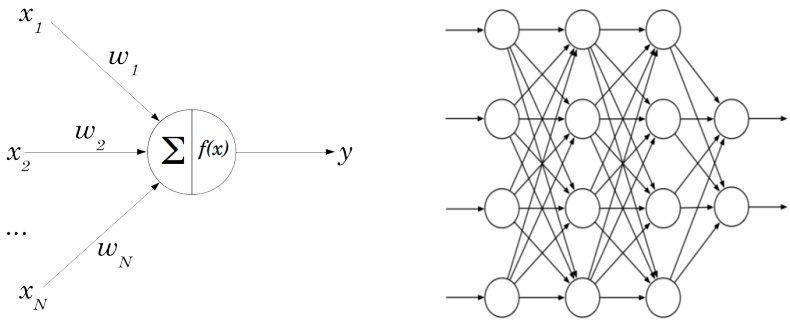
$$F - measure = \frac{2}{1/precision + 1/recall}$$

3.3 Neural Networks

Artificial neural networks are in contrary to common belief, an old technique within the field of machine learning and artificial intelligence. In fact, the first mention of Perceptrons, the elementary building block of today's highly complex neural networks, was proposed as early as in the 1950s by Frank Rosenblatt (Rosenblatt, 1958). It is assumed that the reader possesses knowledge about the basics of neural networks, and therefore the basics are only covered briefly.

The Perceptron seen in 3.2a has n inputs x_1, ... x_n, which is weighted by n weights w_1, ... w_n. The sum of the weighted inputs is given as input to an activation function $f(x)$, which provides a scalar output y. There are many different activation functions to choose from, each having their strengths and weaknesses. Typically, one puts several Perceptrons into layers, providing a network of nodes, as seen in 3.2b. Neural networks are often described to consist of an input, hidden and output layer, as shown in Figure 3.3.

The *input layer* receives the data that the neural net is to process. This data can be encoded in many different ways, depending on the type of input and desired output of the network. Different types of networks, meant for different tasks, is often encoded differently. If the neural network is multi-layered, meaning it has more than an input and output layer, the middle layers are called *hidden layers* as seen in Figure 3.3.



(a) Perceptron (b) Generic feed forward network

Figure 3.2: The Perceptron (a) is the main building block of the artificial neural network (b), which can contain all from a one to hundreds and thousands of Perceptrons.

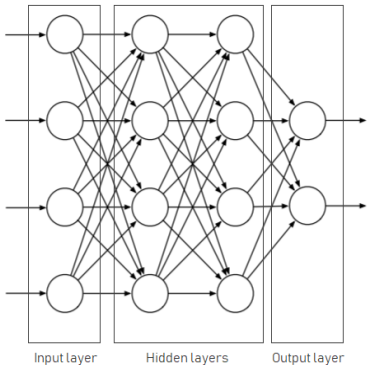


Figure 3.3: The main layers of a neural network

The last layer of the network is called the *output layer*. It is responsible for providing the desired output given the training data. The output layer is highly dependent on what one wishes to accomplish with the neural network. If it is used for binary classification, it could have a single output node, providing a binary prediction. Depending on whether the output node needs to be a probability distribution or not, different activation functions are used. For more complex situations, like in the game of chess, there might be an output node for every possible action with each of the unique pieces on the board, for every position in the $8X8$ grid of the board. The output layer and its activation functions are highly dependent on what the neural net is made to do.

3.4 Convolutional Neural Networks

Convolutional neural networks or CNNs (LeCun et al., 1999), is a type of neural network architecture specially designed for data that has a grid-like topology. Its main focus revolves around data where the spacial properties of the data are what matters the most. This includes the most known application of CNNs, classification of images. Imagine a scenario where you split an image in half down the middle, and you swap the position of the two pieces. That would, unless one presents the image as art, ruin the image and hence the original information within. It is where each piece of information resides in the information space of the data that matters. CNNs are experts on spatial information, and that is why they are so useful in many applications. Three common layers found in almost any convolutional neural networks are the following:

- Convolutional Layers
- Pooling Layers
- Fully connected Layers

The convolution layer

A convolutional layer is responsible for extracting features from the data it is given. When layering convolutional layers on top of each other, one can extract more and more complex features. When using convolutional networks on images, the first layers might find straight lines, the next corners, then shapes like squares

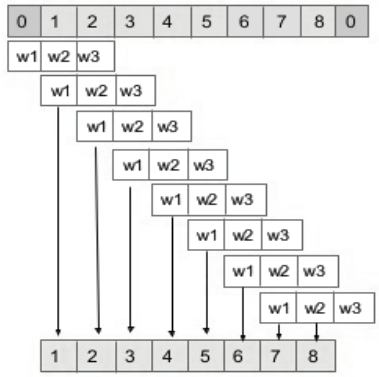


Figure 3.4: Example of 1D convolution with a 1X3 kernel. The kernel slides along the sequence and produces one output per step through the sequence.
Goodfellow et al. (2016)

or circles etc. Complexity is built bottom up, in a hierarchical manner. For one dimensional data like time series or data sequences, this idea is harder to visualise, but the same concept applies.

One dimensional convolution can be defined as

$$s(t) = (w * x)(t) = \sum_{i=-\infty}^{\infty} x(t-i)w(i) \tag{3.2}$$

where x is a one-dimensional data vector of arbitrary length n, and w a weight vector with arbitrary size $< n$. Convolutions can be seen as a sliding window approach, where weights are applied to the data in the window. The same three weights are multiplied with their correlating value in the data-series and the sum of these products are given as output for that convolution, as seen in Figure 3.4

The weights in the kernel are learnable, allowing the network to learn the features of the data. One particularly useful case for one-dimensional convolution is for single-channel data series. Examples of this are time series like single channel Electrocardiography measurements, decibel measurements or light measurements. Similarly to images, which can have multiple channels e.g. red, green and blue, so can single dimension time series. Electrocardiography often has multiple channels, reflecting different modulations of the same raw signal.

The pooling layer

The purpose of the pooling layer is to reduce the spatial resolution of the feature mappings. In other words, it helps to achieve invariance to translations of what the network is trying to detect. In many cases, such as in image recognition or for finely granulated time series, it is often more important to learn about the large structures in the data, rather than each small change within it. Pooling helps create the abstraction from smaller details to larger structures. This is achieved by performing an operation on the sub-sections of the input data, much in the same way as the for the convolution step. One popular example is taking the max value of the pooling area, other options are computing the average or l2 norm.

Fully connected layer

In a convolutional neural network, the convolution layers are responsible for extracting features from the input data. The one or more fully connected layers at the end of the network utilises these features, to extract the global relationship between them. As discussed earlier, the structure of the output layer of this network depends on the task for which it is created to solve. As a simple example, if one is trying to classify images of squares and circles, the output layer could be 2 nodes using softmax as the last activation function. The convolutional layers would be responsible for extracting features like corners, straight lines or curves, and the fully connected layers for deciding whether or not a set of observed features corresponds to a circle or square. Another example could be to design the output layer so that it would provide the most probable moves to be played in a game.

Batch Normalisation

Batch normalization is a technique that has become a common sight in convolutional neural network design. Several advancements in recent years have utilised batch normalization as a part of their architecture, among others within image recognition (He et al., 2016), inception architecture for computer vision (Amodei et al., 2016), and speech recognition (Szegedy et al., 2016).

The normalization technique does something very similar to what is done when normalizing the raw input data, but for the hidden layers in a network. One

term needed to explain the effects of the normalization is *covariate shift*, as introduced by Shimodaira (2000). Covariate shift describes the situation when the distribution of inputs changes, but the conditional distribution of outputs does not. One classic example is a cat-classification neural network, only trained on pictures of black cats. When presented with a different distribution of input, i.e., when trying to classify cats of different colours than black, the network performs badly. In the same way, the downstream layers of a neural network experience covariate shift when the outputs of the upstream layer changes. Batch normalization aims to reduce the effects of this shift, and by doing so reducing the coupling between layers in the network, allowing the later layers to learn more independently from the early layers.

Batch normalization fixes the means and variances of layer inputs. Instead of doing this for the whole dataset at once, which would be very costly, when using stochastic gradient descent each mini-batch B produce the estimates of mean and variance for each activation. The estimates are used to normalise the input of each x_i input vector in the mini-batch. This produces the normalised $\hat{x}_i$ input.

$$\bar{x}_i \longleftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{3.3}$$

This normalised input is used to calculate the output y_i of the batch normalization step. The constant ϵ is added for numerical stability. An important observation to make is that simply normalizing each input of a layer could affect the representational power of the layer. Therefore, two parameters are introduced, to scale and shift the normalised value.

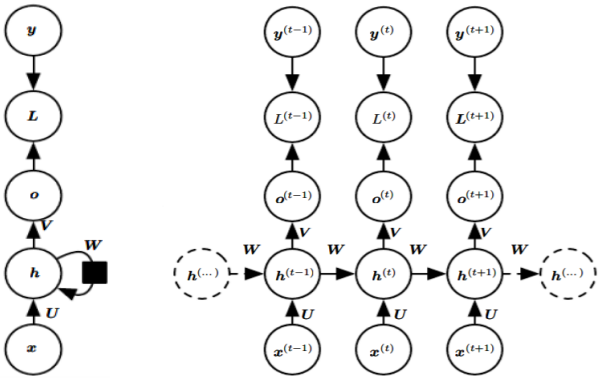
$$y_i \longleftarrow \gamma \bar{x}_i + \beta. \tag{3.4}$$

The two parameters γ and β are learned in the same way as for the network parameters θ. The computation of the gradients can be found in Ioffe and Szegedy (2015). For convolutional networks, all the activations in a mini-batch are normalised jointly over all locations. This ensures that the filters used "behave" the same way for all locations of the input, and a pair of γ, β is learned for each filter.

3.5 Recurrent Neural Networks

Recurrent neural networks (RNNs) belong to a category of neural networks especially designed for sequential data. Sequential data are often thought of as being in the time domain, like daily weather forecasts or data from a sensor reading once every second. However, other types of sequential data like text or DNA-sequences also exists. What these examples have in common is that the property that matters is the order of which the data is in.

A recurrent neural network can be modelled as a series of nodes, where each node corresponds to one part of the learning process, as seen in Figure 3.5a. This representation might seem familiar, as it is in fact not much different from a standard feed-forward neural network. The recurrent nodes in the hidden layers of the network are marked with an h, and it is where the power of the recurrent neural networks resides.



(a) RNN (b) Unrolled RNN

Figure 3.5: Representation of recurrent neural network, where Figure (b) is the unrolled equivalent of Figure (a). Given the input x, the recurrent node h produces the output in node o, given the input and the previous output from h weighted by W. The loss function L is calculated given the label y. Figures from Goodfellow et al. (2016)

The input to a recurrent neural network is given one input at a time, or in other words one step in the sequential the data. A specific input in a sequence is therefore identified by its position or time t in the sequence. The essence of

recurrent networks is that for every given time t, the hidden layers, or more precisely the recurrent nodes in the hidden layers, receive the input $x^{(t)}$ and the output from h for the previous timestep $h^{(t-1)}$, as seen in Figure 3.5b. The input to the node from $h^{(t-1)}$ is weighted by the learnable weights W, which allows the model to learn the same way as for standard feed-forward neural networks. The network can be trained with backpropagation, as for standard feed-forward networks. The details of this can be found in Goodfellow et al. (2016).

One big challenge of RNNs is avoiding the vanishing or exploding gradient problem when the same weights are multiplied numerous times during a sequence of data input. To see this, one can represent the recurrence of the network in a very simplified manner as

$$h^{(t)} = W^\top h^{(t-1)} \tag{3.5}$$

which can be further simplified to

$$h^{(t)} = (W^t)^\top h^{(0)} \tag{3.6}$$

Using the eigendecomposition of $W = Q\lambda Q^\top$ we can further simplify to get

$$h^{(t)} = Q^\top \lambda^t Q h^{(0)} \tag{3.7}$$

Given that the product of $Q^\top Q$ is the identity matrix I, we see that the eigenvalues of the recurrence relation are raised to the power of t. A consequence of this is that values smaller than one will eventually vanish towards zero, and values bigger than one will eventually go towards infinity. In general, this is known as a problem peculiar to recurrent networks, and different approaches exist for reducing its impact.

3.5.1 Gated Recurrent Units

Gated recurrent units or GRUs (Cho et al., 2014) attempt to create gradients that through the duration of a sequence of data neither explode or vanish. The idea was first brought forward as a part of an encoder-decoder network but has been used successfully in several different domains such as network traffic flow

prediction (Fu et al., 2016), recommender systems (Hidasi et al., 2015), and image compression Toderici et al. (2017).

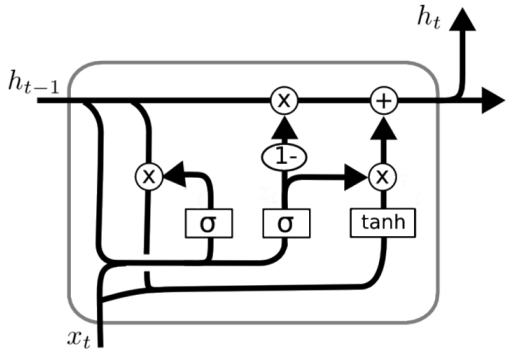


Figure 3.6: Schematic of the gated recurrent unit, where $\bigotimes$ is pointwise multiplication and $\bigoplus$ is pointwise addition. X_t is the input of the sequence data at time t, whereas h_t is the output at time t

The schematic for the GRU-cell is seen in Figure 3.6. Each of the gates in the cell has its own set of weights, allowing the cell to learn when it is important to keep the history, and when it is reasonable to forget it. In addition, dropping information that is irrelevant to the future simplifies the representation. The update gate of the cell allows the model to adjust how much of the history is brought into the next state of the cell, allowing the cell to keep long term information available. This can be thought of as the memory of the cell, which allows producing paths or loops in the cell where the gradient can flow for long sequences of steps.

3.5.2 Long Short-Term Memory Cells

The Long short-term memory cell (Hochreiter and Schmidhuber, 1997) is the second of the two most common types of RNN-cells in use today. The schematic of the Long short-term memory cell (LSTM) unit can be seen in Figure 3.7.

The internal structures of the LSTM-cell are more complex than what of the GRU-cell, and it has some additional features compared to its more simple counterpart. One of those additions is a distinct memory state, which allows for easy

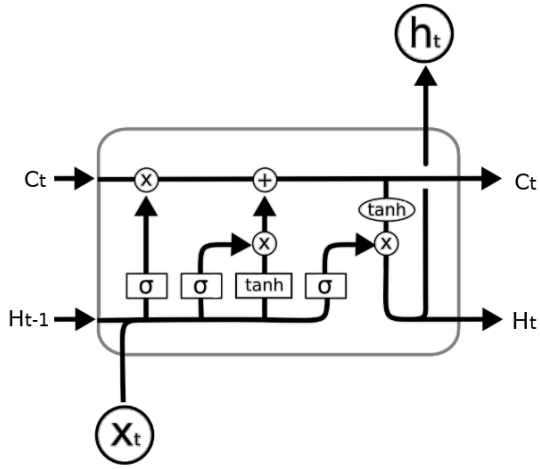


Figure 3.7: Schematic of the LSTM unit, where $\bigotimes$ is pointwise multiplication and $\bigoplus$ is pointwise addition. X_t is the input of the sequence data at time t, whereas h_t is the output.

flow of information throughout the sequence. This memory is represented in the schematic by the top line going through the cell. Three main gates make up the cell, which performs different actions. The first section chooses which information to remove from the cell. The second section chooses which information to add to the cell state, and the third uses the cell state, the previously hidden output h_{t-1} and the input x_t to calculate the new hidden state h_t. An annotated version of Figure 3.7 which visualises this can be seen in Figure 3.8.

3.6 Random Forests

Random forests are an ensemble learning method that uses multiple decision trees (Ho, 1998) in order to make a decision based on some function of each trees output. Employing multiple classifiers to address the problem space, or hypothesis space, results in increased robustness against overfitting, an important goal in the original proposal of Random forests.

The underlying mechanism for Random forests, decision trees, is presented in Section 3.6.1. Ensemble learning is defined in Section 3.6.2, before presenting the

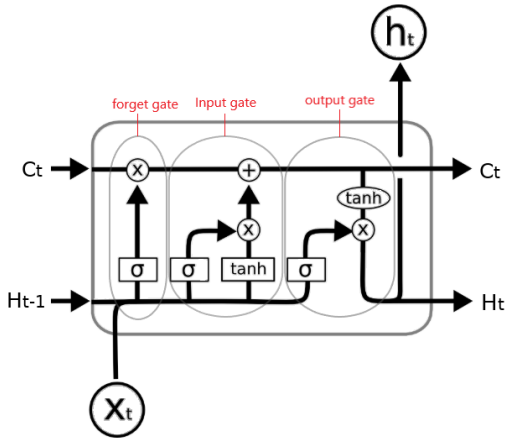


Figure 3.8: Schematic seen in Figure 3.7, annotated with the functionality of each section of the cell.

process of ensembling decision trees and the details of Random forests in Section 3.6.3. Feature importance in relation to Random forests is discussed in Section 3.6.4, which is often used for assessing the relevance of features in a dataset.

For further clarification, random forests and decision trees are usable on both regression and classification tasks, however, the following presentation is limited to the classification case only.

3.6.1 Decision Trees

A decision tree is formally a function, where the output is given by inducing a series of tests on a given input, following the ID3 algorithm (Quinlan, 1986). Decision trees are known as on the simplest yet most successful forms of machine learning (Russell and Norvig, 2016). An example of a Boolean decision tree is given in Figure 3.9.

Following the algorithmic approach of ID3 (Quinlan, 1986), a decision tree is built by dividing the training set into multiple subsets. The procedure of splitting a data set for classification tasks is guided by the information gain, or in other words the change entropy for a split. This allows the decision tree to carefully extract meaningful rules, as seen in Figure 3.9, for correctly labelling unseen

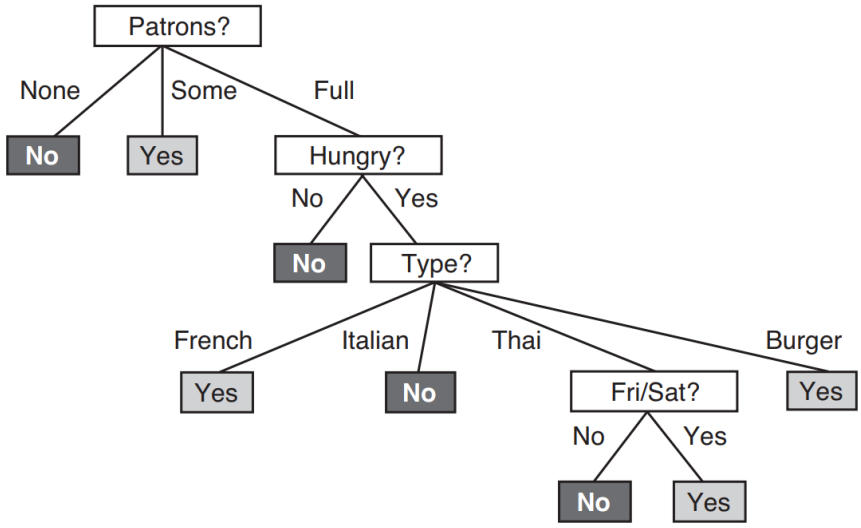


Figure 3.9: A visual example of a how Boolean decision tree work, in the search for when and where to eat (Russell and Norvig, 2016)

data. Looking at Figure 3.10, it becomes clear that splitting at *type* (option a), brings the tree no closer at classifying whether to eat. However, splitting at *patrons* (option b), has a far better information gain, which highlights how and why entropy is of key importance for creating decision trees.

3.6.2 Ensemble Learning

Ensemble learning methods gather several individual classifiers in an ensemble, whose predictions are combined when classifying instances (Opitz and Maclin, 1999). There is no practical restrictions on which classifiers that are usable or not for ensemble learning, implying that there is a wide variety of approaches and methods available.

Given an ensemble of five different individual classifiers, while combining predictions by the majority vote, only three classifiers need to classify correctly for the ensemble to predict correctly. Another popular analogy is that ensemble learning combines multiple weak classifiers, into a joint strong classifier. This provides a strong rationale for applying ensemble learning when a single classifier provides

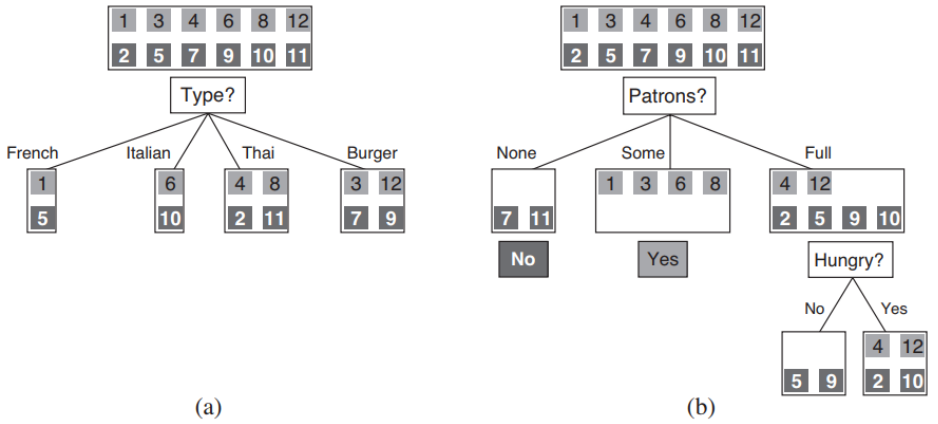


Figure 3.10: A visual example of a decision tree can benefit from splitting data depending on information gain, in the search for when and where to eat (Russell and Norvig, 2016)

inadequate results by itself.

3.6.3 Ensembling Decision Trees

Random forests are by definition an ensemble method, where multiple decision trees are combined into a joint *Random forest*. Multiple approaches within ensemble learning are proposed in the literature, but for this context, only random forests are of interest.

The motivation for introducing random forests, or the random subspace method as originally named, was to maintain the high accuracy of decision trees while fortifying the ability to generalise (Ho, 1998). The method is especially resistant against overfitting, at least when compared to single decision trees.

Random forests consist of multiple trees created systematically by selecting subsets of the training set (Ho, 1998), into a final *forest*. That is, multiple trees are constructed where each tree is specifically trained on a given subset. A visual example is given in Figure 3.11.

Input is given to the random forest, which then passes a random subset onwards to each decision tree. Each tree performs their respective computations, and

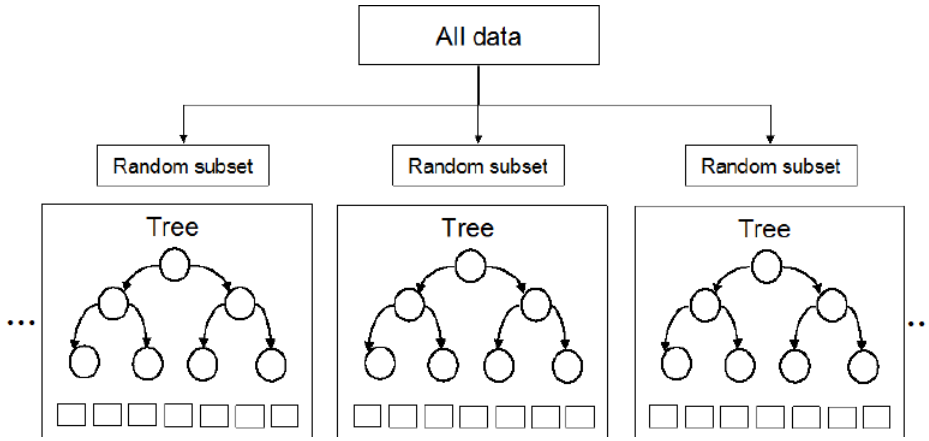


Figure 3.11: A visual example of how random forests are structured (Isied and Tamimi, 2015). The dataset is split into N subsets, and a tree is constructed and train on each subset of the data.

each provides its own output. The random forest algorithm then collects those outputs and combines them into a single output. How the outputs are combined into a single value often varies depending on the situation, and a multitude of methods are available for this problem. As noted in Section 3.6.2 majority vote is a common approach.

3.6.4 Feature Importance

As briefly discussed in Section 3.6.1 information gain is a key concept for random forests. Having a clear indication of which features are contributing to effective classification allows random forests to trivially tell which features are the most relevant. Unlike, for example, artificial neural networks, random forests and decision trees are built by effective consideration of how valuable a feature is for classification.

A visualisation of a typical example of how Scikit-learn (Pedregosa et al., 2011) presents feature importance is given in Figure 3.12. Feature 1, 2 and 0 are the major contributors, while the remaining features have less importance for classification.

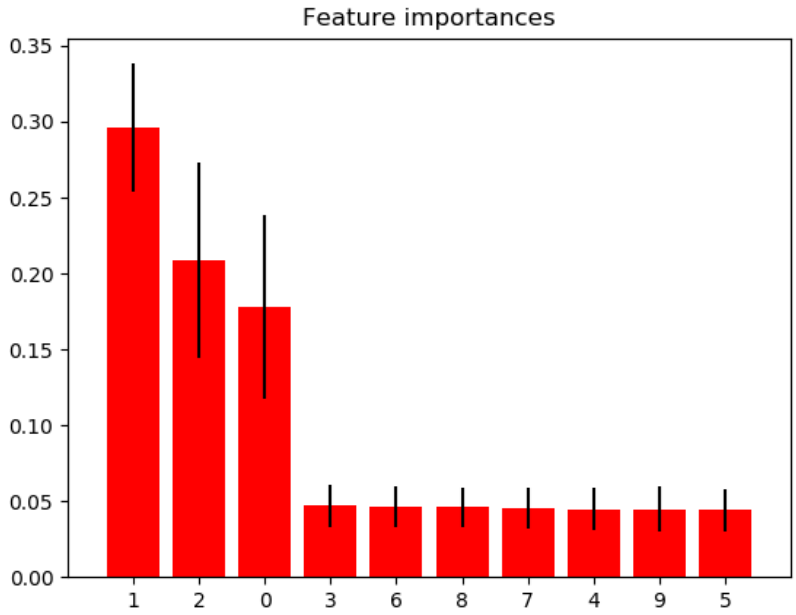


Figure 3.12: A visual example of how feature importance's are ranked for an artificial classification task (Pedregosa et al., 2011)

For decision trees feature importances are derived trivially from the previously discussed topic of information gain and entropy. Random forests combine multiple decision trees, which adds a layer of complexity as feature importance is now computed across all trees. Scikit-learn (Pedregosa et al., 2011) computes feature importances by computing the mean importance for each feature across every tree, as presented in Equation 3.8. For clarification, different approaches exist for computing the joint importance for ensemble models, however, the mean method is the one implemented by Scikit-learn on forest architectures. In conclusion, random forests and decision trees are suitable for devising metrics on feature importance and the relevance of each feature for effective classification.

The equation for computing feature importance when using ensemble models using the mean of each feature is given in Equation 3.8 below,

$$[H]FI_i = \frac{1}{n}\left(\sum_{j=1}^{n} x_{i,j}\right), \tag{3.8}$$

where FI_i the is feature importance for feature i, n the number of features and $x_{i,j}$ the feature importance for feature i and decision tree j.

3.7 K-fold Cross Validation

Cross-validation is applied during the final testing of the model, which in a better way than normal sampling describes how well the model generalises. The key element of the procedure is to partition the data into k distinct parts or folds. Using $k = 10$ would result in a 10-fold cross-validation.

The data is split into k distinct folds, and in an iterative fashion, each fold is used once as test data while the remaining $k - 1$ folds are used as training data. This implies each partition is used as a test fold 1 time, and as a training fold $k - 1$ times. The evaluation metrics are retained for each iteration and summarised in a joint metric once the cross-validation is finished. Taking the mean of the evaluation metrics is a trivial approach for creating the final performance metrics. Figure 3.7 visualises how a 5-fold cross validation partitions data in an iterative fashion.

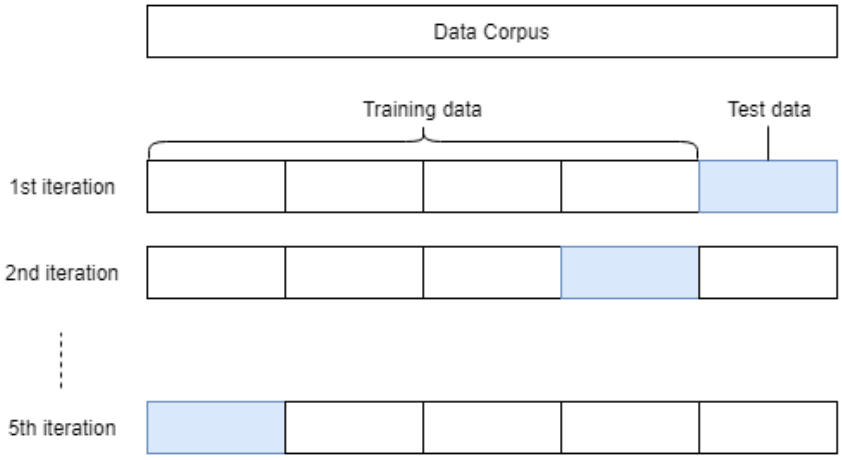


Figure 3.13: Visualization of a 5-fold cross-validation procedure. The data is split into 5 equally sized subsets, and for each iteration, a new subset is used as the test set for the model.

A practical implication of using cross-validation where test data is routinely exchanged is that any scaling of data has to occur every iteration. Scaling test data together with training data can result in an optimistic evaluation, as scaling the entire corpus would also consider the distribution of the test data. A normal approach is to scale the training data, retain the parameters used for scaling, and then to scale to test data with the same parameters. The key observation here is that the scaling parameters must only be based on the training data. This ensures testing is done without any prior knowledge of the test data.

Chapter 4

State of the Art

The following chapter presents the state of the art approaches related to sleep-disordered breathing recognition. Section 4.1 presents the current literature on recognition of sleep-disordered breathing, Section 4.2 explores the presents theory on combining recurrence and convolution, while Section 4.3 presents state of the art techniques for counteracting class imbalance in the research field.

4.1 Sleep-Disordered Breathing

There is a wide variety of approaches for detection of sleep-disordered breathing (SDB). A large number of variations is discussed in the literature, where both signals and classification approach varies greatly (Uddin et al., 2018). A review of Doppler-based radar systems for OSA detection (Tran et al., 2019) shows that the predictions are often made using either time-frequency analysis, numerical analysis or machine learning.

The signals used for detection of sleep apnea range from raw PSG signals to features extracted from bed mattress sensors. As the term sleep apnea is a collection of several diseases, exactly what is being classified often differs between studies.

For further reference, note that it is difficult to compare results between publications due to several reasons. Variations in which signals are used is a key factor.

In addition, each study uses different datasets. The occurrences of apnea and hypopnea might differ largely, and highlights why some datasets are either easier or more difficult to predict. At the same time, the target result of the research might differ. Examples of typical approaches are predicting whether an actual subject or patient has apnea, if apnea is found in a given epoch or estimation of the apnea-hypopnea index.

4.1.1 Doppler Based Radar Systems

Using radar for measuring vital signs is a well-tested method, which has shown good results and a high correlation to the true signal when compared to PSG. Although not as precise, or with the ability to record the exact same signals, the use of radar data has shown promising results (Tran et al., 2019). The main benefit of using radar is the cost efficiency and the non-intrusive approach for gathering data, which also scales better than PSG with regards to the availability of equipment and cost of operation.

SleepMinder is a non-intrusive vital signs monitor, which achieved a correlation of 91% to PSG when estimating the AHI (Zaffaroni et al., 2009). The recordings were tagged with movement flags by a movement detector algorithm, which was used for later numerical analysis. By extracting features using signal analysis techniques the system was able to detect sleep-disordered breathing events when overall body breathing effort declined past a given threshold. To compute the AHI score, a sleep-wake analysis was performed making it possible to compute SDB events per hour of actual sleep. The accuracy of SleepMinder was later empirically verified in Zaffaroni et al. (2013) and Savage et al. (2016), which reported a correlation of 90% to PSG and overall accuracy of 85.8%, respectively.

A similar approach to what was used in SleepMinder was later proposed by Kagawa et al. (2013). Featuring a system with two Doppler-based radars measuring abdomen and chest separately, a correlation of 98% to RDI (respiratory disturbance index) was achieved. The RDI indicates the number of events per hour of recording, and not per hour of sleep which SleepMinder used. As for SleepMinder, domain knowledge was used for numerical analysis and feature engineering. Both studies used respiration analysis, utilizing a drop in amplitude for detecting apnea events.

4.1.2 PSG Based Systems

Using an ensemble architecture, consisting of sparse auto-encoders, SVM, ANN and HMM, Li et al. (2018b) attained an 85.0% classification accuracy with a segment size of 60 seconds. ECG was used as the data signal, which was preprocessed and interpolated into 100 data point segments describing the respiration rate (RR). That is, a segment of 100 data points of RR equated to 60 seconds of ECG recordings. The architecture of the classification model is visualised in Figure 4.1.

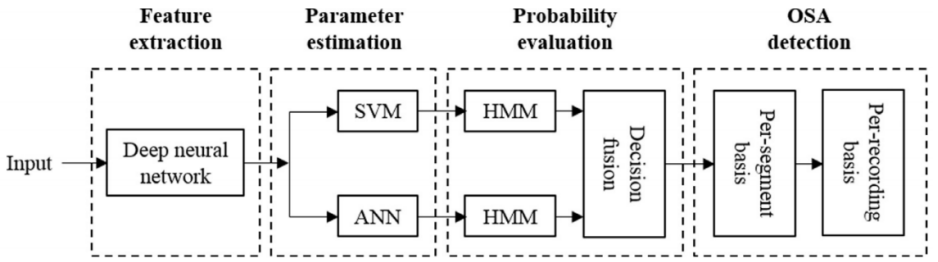


Figure 4.1: The architecture of the proposed model in Li et al. (2018b). The model uses a deep neural network to extract features from the data, ensemble learning to create predictions and HMM to model temporal dependencies between epochs.

As explained in Li et al. (2018b), the auto-encoder is used to effectively represent the data internally for the model. The SVM and ANN are used to exploit the power of ensemble learning, while the HMM preserves the temporal connection between the data points. The approach outcompeted previous research considering research using the same data corpus, but it was noted that classifying different cases of sleep apnea remains an unsolved challenge.

A convolutional neural network model was proposed by Dey et al. (2018), where the presence of apnea in epochs was predicted. Attaining an accuracy of 98.9%, sensitivity of 97.8% and specificity of 99.2%, the CNN delivers some of the most promising results seen as of today. ECG was used as input, without any handcrafted preprocessing, which deviates from most other research where heavy feature engineering is applied.

Convolutional neural networks have also been used for multi-label classification including hypopnea. Without any specific feature engineering, Urtnasan et al.

(2018b) attained an accuracy of 90.8%, with precision and recall at both 87.0% using ECG as the source of the signals. Evidently, including hypopnea increases the difficulty of the problem. Although this is not directly comparable to Dey et al. (2018), Urtnasan et al. (2018b) seems to be outperformed by its stronger performing equivalent.

The same team behind the SDB classification problem using CNN (Urtnasan et al., 2018b) later published a new study, but this time using RNN (Urtnasan et al., 2018a). This approach, on the other hand, attained a precision and recall of respectively 97.0% and 96.0%. ECG, without any domain-specific preprocessing, were also for this study used as input.

A novel hand-made signal processing technique, *tunable-Q factor wavelet transform* (TQWT), used alongside RUSBoost (Seiffert et al., 2010), obtained strong results in the detection of OSA. RUSBoost is a boosting algorithm making it suitable for training on imbalanced data, which is a known problem area for sleep apnea. The classifier achieved an accuracy of 91.9%, sensitivity of 90.4% and specificity of 92.7%.

4.1.3 Non-radar Based Systems

Many of the suggested home-testing methods for detecting sleep apnea in the literature uses sensors and devises other than radar. Despite using different sources of data, these approaches still provide valuable information to what features they employed, and to which degree these were relevant. An important aspect of their success is how precisely the technology can read the vital signs of the subject.

A microbend fibre optic cable sensor as part of a bed mattress system was proposed to monitor vital signs by Sadek et al. (2018). Although the research focused on the accuracy of reading vital signs, an AHI score was computed to further validate and test clinical usability. The system achieved a specificity of 85.9% and sensitivity of 24.2% using ApneaLink (Erman et al., 2007) as the target when predicting the number of apnea events. As for similar non-machine learning approaches, handcrafted features were essential for classification. Using respiratory signal extracted from the fibre optic cable, an apnea is tagged if the respiration signal drops sufficiently over a defined period.

Technologically different, but in the form of a bed mattress, Davidovich et al. (2016) proposed a sleep apnea detection system using a piezo-electric sensor.

Similar to several of the other studies mentioned, apnea events were found by respiration analysis using handcrafted features. The goal was to place the subjects in either of two categories depending on whether their AHI score is above 15, i.e. either moderate/severe or mild/none sleep apnea. Using measurements from PSG as the ground truth, the bed mattress achieved a sensitivity of 88% and specificity of 89% for the moderate-to-severe subjects, with an overall correlation of 0.86.

4.1.4 Overview

In addition to the research presented in the previous section, there are numerous papers published in recent years. The papers presented were chosen because they showed the most promising result for that specific classifier. As there are countless variants of the granularity and classes predicted, as well as the results presented in the literature, it is hard to identify which approach is the most successful. Therefore, Table 4.1 presents the overview of methods and metadata used among the articles considered. Some publications report results for multiple approaches, including different signals and different classifiers. For those cases, only the best performing approach is presented in Table 4.1.

Study	Input	Method	Event	Granularity
Zaffaroni et al. (2009)	SM[a]	FE[b]	AHI[c]	Recording
Zaffaroni et al. (2013)	SM	FE	AHI	Recording
Savage et al. (2016)	SM	FE	AHI	Recording
Li et al. (2018b)	ECG	Ensemble ML	O[d]	60s epoch
Dey et al. (2018)	ECG	CNN	O	60s epoch
Urtnasan et al. (2018b)	ECG	CNN	O/H[e]	10s epoch
Urtnasan et al. (2018a)	ECG	RNN	A/H[f]	10s epoch
Van Steenkiste et al. (2018a)	EDR	RNN	O/C/H[g]	30s epoch
Janbakhshi and Shamsollahi (2018)	EDR	ANN	O	60s epoch
Cheng et al. (2017)	ECG	RNN	O	Recording
Hassan and Haque (2017)	ECG	RUSBoost	O	60s epoch
Sharma and Sharma (2016)	ECG	SVM	A	Recording
Gutiérrez-Tobal et al. (2019)	Oximetry	AB-LDA[h]	AH-class[i]	Recording
Tripathy (2018)	HRV/EDR	KELM	A	60s epoch
Haidar et al. (2017)	Airflow	CNN	O	30s epoch
Avcı and Akbaş (2015)	Airflow	Random Forest	A	60s epoch
Martín-González et al. (2017)	HRV	QDA[j]	A	300s epoch
Bianchi et al. (2014)	Thoracic	FE	O	30s epoch
Koley and Dey (2013b)	Airflow	SVM	A/H	10s epoch
Koley and Dey (2013a)	Airflow	SVM	A/H	8s epoch
Sadek et al. (2018)	BM[k]	FE	AHI	10s epoch
Davidovich et al. (2016)	BM	FE	AHI	3s epoch

Table 4.1: Overview of considered research within sleep-disordered breathing recognition.

[a] SleepMinder
[b] Feature Engineering
[c] Apnea-Hypopnea Index
[d] Obstructive Apnea
[e] Obstructive Apnea and Hypopnea
[f] Apnea and Hypopnea
[g] Obstructive and Central apnea and Hypopnea
[h] AdaBoost - Linear Discriminant Analysis
[i] Severity of Apnea-Hypopnea diagnosis
[j] Quadratic Discriminant Analysis
[k] Bed Matress

4.2 Combining Convolution and Recurrence

In addition to the current state of the art research, there are multiple other research areas from which one can get inspiration. One of the more closely related areas to detecting sleep disorders is predicting sleep stages, as seen in Table 2.1, which is often used to perform sleep quality analysis. One particularly interesting method is the combination of a convolutional neural network and a recurrent neural network (Zhao et al., 2017). The model predicts the current sleep stage of the subject using the reflections of radio frequency signals, resulting in a comparable data source to that of Somnofy as seen in Section 2.3.2. The CNN is used to distinguish between the different sleep stages, but a challenge arises due to the difficulties separating deep and light sleep. Adding a recurrent layer after the convolutional layers improved the overall accuracy with 12.8 %, and the improved the model's ability to distinguish between light and deep sleep with 23.5 %.

This provides strong support for the general strategy of extracting and classifying spatial or short term features with convolutional layers, and longer term or temporal features with recurrent layers. However, the efficiency of these methods depends on a temporal dependency between the states of the time series. The joint architecture is seen in other domains as well, such as gesture recognition in video (Pigou et al., 2018) and multi-label image classification (Wang et al., 2016).

4.2.1 Choice of Model for Recurrent Neural Layers

There are multiple alternatives for which type of recurrent neural network to use, whereas the two most commonly adapted versions are LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Cho et al., 2014). Both have shown good performance compared to the general RNN-model, and are often performing similarly compared to each other. Several studies have compared the two variants in areas such as natural language processing (Yin et al., 2017), music, and speech modelling (Chung et al., 2014), and traffic flow prediction (Li et al., 2018a). Few have shown a significant difference between them, but with a more powerful representational power, LSTM seems to be slightly ahead on performance in general compared to GRU. While using LSTM might increase performance to some degree, it comes with the cost of an increased need for computational power.

4.3 Imbalanced Datasets

A patient diagnosed with a severe degree of sleep apnea has 30 or more events per hour, where each event typically lasts in between 10 to 20 seconds. There are different forms of sleep apnea, which in this context is restricted to OSA, MSA, and CSA, where OSA is the most prevalent as described in Section 2.2. In addition, hypopnea is often present in patients with other forms of apnea. However, even for subjects with severe apnea, the majority of the data consists of healthy periods. As a result of this, sleep apnea-data is heavily imbalanced between apnea, hypopnea, and healthy data. As noted in Section 3.1, an imbalance can have a significant detrimental effect on accuracy. This section presents techniques for dealing with imbalance, alongside a detailed presentation of the unbalancing techniques used in literature for sleep-disordered breathing recognition.

A detailed description of the dataset used in this thesis is presented in Chapter 5, which further quantifies the degree of imbalance for the datasets used in this thesis.

4.3.1 Class Imbalance in Sleep Disorder Detection

Table 4.2 gives a detailed overview of SDB research that describes what approach is taken for working with imbalanced data. A substantial amount of the available research does not describe their method in detail, which makes the overview subject to some interpretation.

Five cases of undersampling are observed, where four are sampling randomly from the majority class, while the last study uses RUSBoost. As seen in Section 4.1.2, RUSBoost removes samples from the majority class while employing boosting, which trains multiple classifiers on different subsets of data before combining the classifiers in a predetermined way to create a single output. One single study employs oversampling by utilising bootstrapping, where samples are selected randomly from the majority class(es). Two studies used cost-based learning.

As observed in Table 4.2, class imbalances problems remains untreated in a majority of the studies, or without a proper focus on imbalance, referring to the lacking description of the method used. The degree of impact the technique used for balancing data has on performance is difficult to quantify, as comparing papers and approaches is non-trivial. Some studies also have an abundance of

Study	Technique	Comments
Zaffaroni et al. (2009)	None	-
Zaffaroni et al. (2013)	None	-
Savage et al. (2016)	None	-
Li et al. (2018b)	None	-
Dey et al. (2018)	None	-
Urtnasan et al. (2018b)	Undersampling	Random
Urtnasan et al. (2018a)	None	-
Van Steenkiste et al. (2018b)	Undersampling	Random
Van Steenkiste et al. (2018a)	Oversampling	Balanced bootstrapping
Janbakhshi and Shamsollahi (2018)	None	-
Cheng et al. (2017)	None	-
Hassan and Haque (2017)	Undersampling	RUSBoost
Sharma and Sharma (2016)	Undersampling	Random
Gutiérrez-Tobal et al. (2019)	None	-
Tripathy (2018)	None	-
Haidar et al. (2017)	Undersampling	Random
Avcı and Akbaş (2015)	None	-
Martín-González et al. (2017)	None	-
Bianchi et al. (2014)	None	-
Koley and Dey (2013b)	Cost-based learning	-
Koley and Dey (2013a)	Cost-based learning	-
Sadek et al. (2018)	None	-
Davidovich et al. (2016)	None	-

Table 4.2: Overview of class imbalance techniques employed within sleep-disordered breathing recognition.

data using larger datasets from publicly available studies, where data is sampled randomly for each label. This makes the existing class imbalance in the dataset trivial as random sampling is sufficient for providing enough training data for the model.

4.3.2 Data Balancing Techniques

Buda et al. (2018) performed a systematic study of using convolutional neural networks with imbalanced datasets. The study recommended a wide range of suggestions for approaching the imbalanced data. A key observation is that the impact of the imbalance depends on the distribution among the examples, and not only the ratios between the majority and minority classes. Severe imbalance can occur even when the ratios between the majority and minority classes are relatively low.

The study indicates that oversampling generally performs better in cases with multi-class problems. Thresholding, a common method for adjusting the decision threshold of predictions, was shown to improve the performance of oversampling in some cases. Another key observation from Buda et al. (2018) is that oversampling performs optimally when the imbalance is performed such that every class has the exact same size. Finally, oversampling is not observed to cause overfitting when used alongside convolutional neural networks, which is in contrast to other classical machine learning approaches, where overfitting is easily seen when creating data with a narrow distribution. Undersampling was shown to perform equivalently to oversampling for extreme ratios of imbalance and has the added benefit of reducing the size and hence computational power needed to train the model. A major drawback of undersampling is that by reducing the population of cases, information is removed that could be utilised by the algorithm.

A different study using CNN to diagnose breast cancer in images (Reza and Ma, 2018) reported a benefit using oversampling techniques when comparing imbalance methods on two different data corpora. The best performing approaches in this study were basic oversampling and SMOTE (Chawla et al., 2002). Undersampling performed the worst. However, there were large variations between the approaches in terms of accuracy measures between the two datasets, which makes it hard to generalise which method outperforms the others.

SMOTE stands for *Synthetic Minority Over-sampling Technique* and is a hybrid approach between oversampling and undersampling. Briefly explained, SMOTE

oversamples the minority class while undersampling the majority class. Oversampling is done by synthetically creating minority examples, where methods such as K-Nearest Neighbors (KNN) can be used to create the new data points. The goal behind synthetic examples is to increase the distribution. The idea behind an approach like SMOTE is that using a non-random heuristic for either under or oversampling leads to better choices of samples by using information in the data to sample the most optimal set of data points. Other methods available for generating data is Generative adversarial networks (GAN), that has been tested as a technique for oversampling minority classes in fault detection and diagnosis (Suh et al., 2019). The GAN creates synthetic examples by using information from the known examples and is in many ways similar to using KNN in SMOTE. Using GANs showed a clear tendency to improve accuracy, but not more than 2% at the most.

On that remark, it seems difficult to point out a single imbalance technique as superior. It does, however, seem likely that variables such as imbalance ratios and distribution of data is heavily affecting the aspect of imbalance, and therefore which methods are the most suitable to a given dataset.

Chapter 5

Data

This chapter presents the data corpus used for training, validation, and testing. VitalThings conducted two data collection sessions, hereafter named Bergen and Colosseum. All sessions were recorded with PSG, a Somnofy mounted on a wall above the bed and a Somnofy placed on a nightstand next to the bed. So for both sessions, there are three sources of data, whereas PSG is used as the target. Section 5.1 presents metadata related to the Bergen trial, while Section 5.2 presents metadata related to the Colosseum trial. Section 5.3 presents key features from the trials.

5.1 Bergen

Recorded in 2017, The Bergen dataset is a sample of healthy persons meeting a set of predetermined inclusion and exclusion criteria. What follows is a detailed description of the patients included in the trial. Table 5.1 presents metadata about the patients. A relatively young and uniformly distributed population, in comparison to the official yearbook of Statistics Norway 2013 (SSB, 2013). The yearbook was discontinued in 2013. The largest percentage-wise deviation is the weight of males with a difference of $2.9kg$. The yearbook does unfortunately not present any standard deviations. A visualisation of the distribution for birth year, height and weight is presented, respectively, in Figure 5.1, 5.2 and 5.3.

Metric	Female	Male	Total
Gender	25	18	43
Age ($year$)	25.1 ± 4.1	32.6 ± 9.2	28.2 ± 7.7
Height (cm)	167.9 ± 6.7	180.4 ± 6.9	173.1 ± 9.2
Weight (kg)	62.4 ± 8.0	76.8 ± 10.2	68.5 ± 11.5
BMI (kg/m^2)	22.1 ± 2.1	23.5 ± 2.4	22.7 ± 2.3

Table 5.1: Metadata on the patients recorded from Bergen 2017

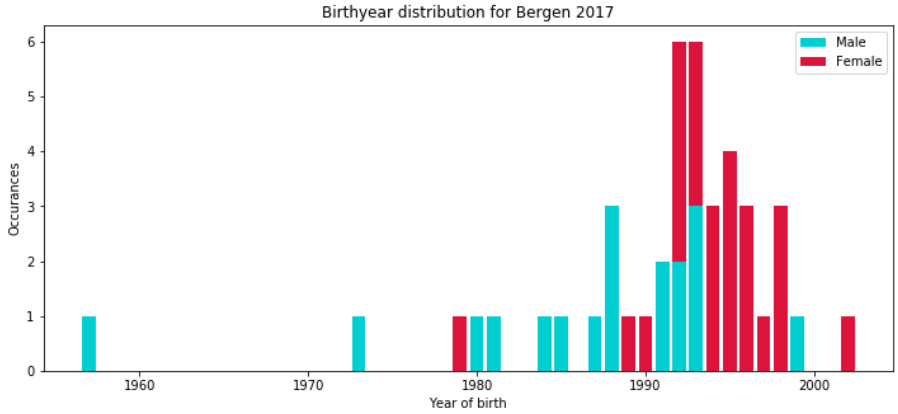


Figure 5.1: Distribution of birthyear for Bergen 2017

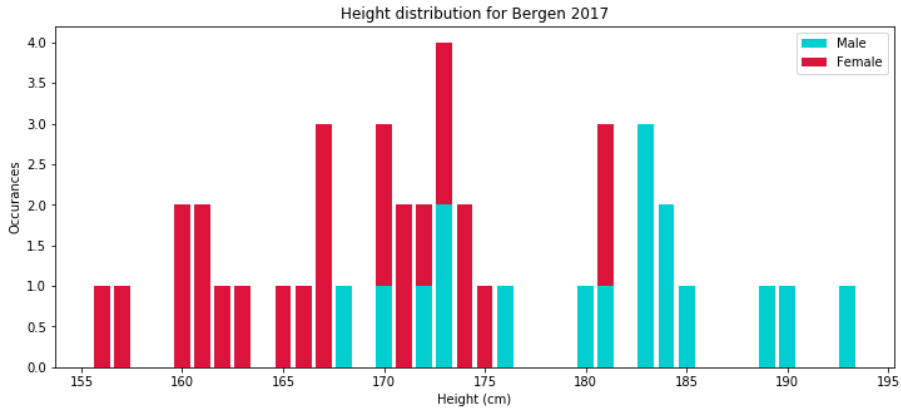


Figure 5.2: Distribution of height for Bergen 2017

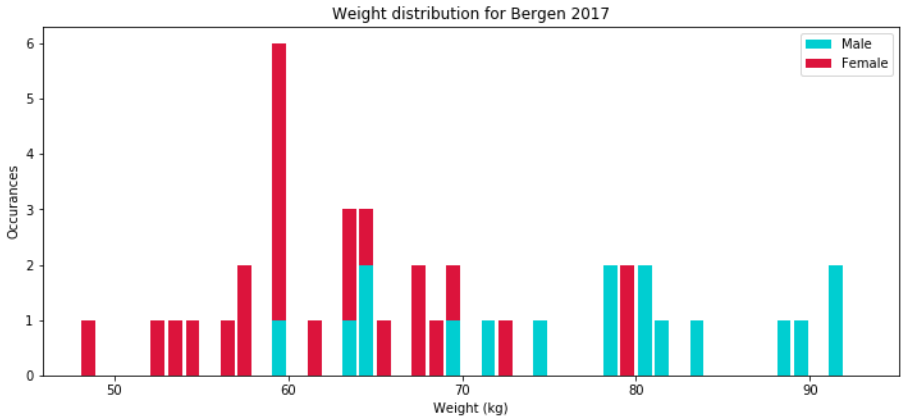


Figure 5.3: Distribution of weight for Bergen 2017

Among the population from Bergen, few or none were expected to have any sleep-related diseases. This comes from the inclusion and exclusion criteria used in the study. This is reflected in the analysis performed on the manual PSG diagnosis-data, where the mean and standard deviation events per person per night is presented in Table 5.2. Although the presence of obstructive apnea and hypopnea events are clear, the standard deviations are relatively high for all categories.

Metric	**Female**	**Male**	**Total**
Body Event	0.0	0.0	0.0
Central Apnea	0.3 ± 0.9	1.1 ± 1.5	0.6 ± 1.2
Flow Limitation	0.6 ± 0.7	0.3 ± 0.5	0.5 ± 0.6
Hypopnea	14.8 ± 12.4	29.3 ± 17.2	21.0 ± 16.3
Mixed Apnea	0.1 ± 0.3	0.1 ± 0.3	0.1 ± 0.3
Obstructive Apnea	2.8 ± 3.8	5.3 ± 6.4	3.9 ± 5.2
RERA	0.0	0.0	0.0
Events (total)	18.6 ± 11.5	36.1 ± 22.1	26.1 ± 19.0

Table 5.2: Overview of sleep disease related events per night per person recorded from Bergen 2017

5.2 Colosseum

The Colosseum dataset was recorded during the summer of 2018 and was expected to have a healthy population meeting a set of inclusion and exclusion criteria as for Bergen. A total of 53 unique nights were recorded, where 11 of those nights were later tagged as either *diagnosed*, *possible diagnosis* or *sick*. None of the nights was tagged for the Bergen trial. It is unknown whether they simply are unlabelled, or didn't qualify for a diagnosis.

The patients follow the expected mean height and weight, referring to the Norwegian average recorded by Statistics Norway in 2013 (SSB, 2013), as shown in Table 5.3. The largest deviation, as for Bergen, is still the male weight. The deviation is even larger for Colosseum but this is likely related to a slightly older population. It is seen in the literature that obesity and overweight increases the likelihood of sleep-related diseases (Resta et al., 2001), such as sleep apnea. The distribution of birth year, height and weight for Colosseum 2018 is also visualised in Figure 5.4, 5.5 and 5.6.

Metric	**Female**	**Male**	**Total**
Gender	24	29	53
Age ($year$)	36.4 ± 12.9	36.9 ± 12.6	36.7 ± 12.8
Height (cm)	166.9 ± 4.9	180.1 ± 6.3	174.2 ± 8.7
Weight (kg)	64.5 ± 10.5	80.0 ± 9.5	73.0 ± 12.6
BMI (kg/m^2)	23.1 ± 3.5	24.6 ± 2.2	23.9 ± 3.0

Table 5.3: Metadata on the patients recorded from Colosseum 2018

An overview of sleep-related events is presented in Table 5.4. The mean occurrence of events is relatively higher than for Bergen in some categories and is possibly correlated to the Colosseum population being slightly heavier. One feature of the distribution worth mentioning is that the male standard deviation is abruptly large for several categories. This implies that a few male patients have severe sleep-disordered breathing, which can explain the deviance between the Colosseum and Bergen trial.

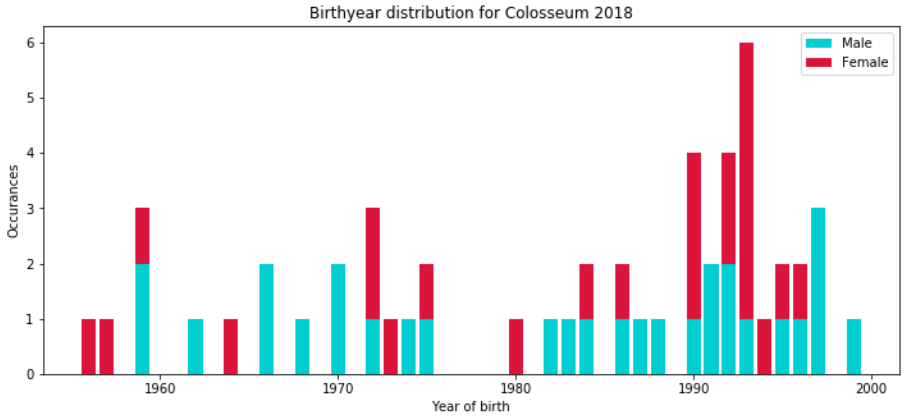


Figure 5.4: Distribution of birthyear for Colosseum 2018

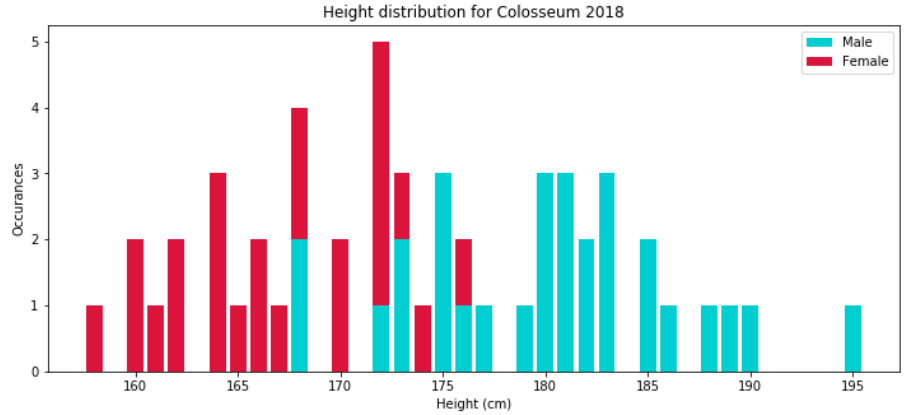


Figure 5.5: Distribution of height for Colosseum 2018

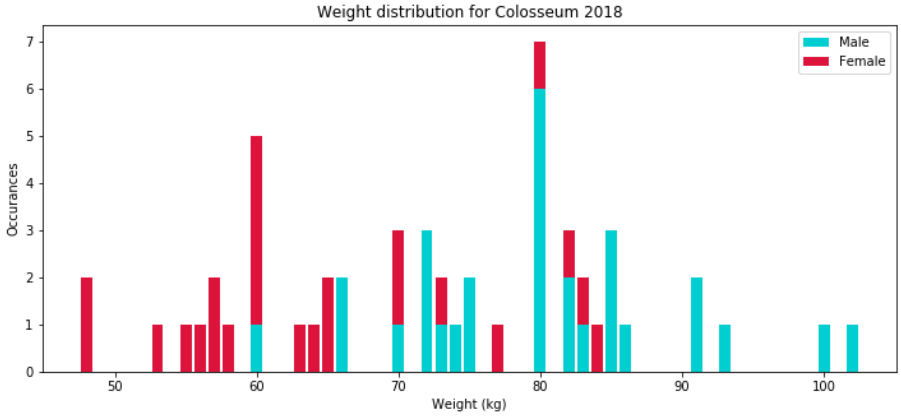


Figure 5.6: Distribution of weight for Colosseum 2018

Metric	**Female**	**Male**	**Total**
Body Event	0.6 ± 1.1	1.3 ± 1.8	1.0 ± 1.6
Central Apnea	0.2 ± 0.4	2.4 ± 7.3	1.4 ± 5.6
Flow Limitation	0.2 ± 0.4	0.1 ± 0.3	0.2 ± 0.4
Hypopnea	16.9 ± 20.8	46.1 ± 45.8	33.2 ± 39.6
Mixed Apnea	0.0 ± 0.2	3.2 ± 14.3	1.8 ± 10.8
Obstructive Apnea	0.9 ± 1.8	6.1 ± 24.7	3.8 ± 18.7
RERA	0.7 ± 0.9	1.2 ± 2.1	1.0 ± 1.7
Events (total)	19.4 ± 21.9	60.5 ± 84.0	42.3 ± 67.6

Table 5.4: Overview of sleep disease related events per night per person recorded from Colosseum 2018

5.3 Data Features

A described in Section 2.3.2, Somnofy has multiple signals accessible in their recordings. Some of those signals are feature engineered, either by VitalThings themselves or by the chip manufacturer Novelda. The following section describes which signals and features that are available for use when recognising sleep-disordered breathing, except for PSG which already was presented in Section 2.3.1. Recall that both of the Somnofy trials is also supplemented with a PSG recording scored by an authorised physician. That is, all Somnofy recordings have a ground truth, or gold standard, through PSG recordings. The quality of a Somnofy signal is therefore comparable to that of PSG. Recall that SDB is short for sleep-disordered breathing, which is a term frequently used for the following section.

One of the most relevant vital signs for sleep-disordered breathing is respiration rate. Fluctuations in airflow, alongside desaturation, are the key metrics for scoring apnea and hypopnea, as presented in Section 2.2. VitalThings has by feature engineering created a signal which measures the distance to the chest of a subject. This allows for precise measurement of inhalation and exhalation, which can be used to create a feature for the respiration rate. The respiration rate over time is called the *Somnofy Respiration Curve*. A thoracic PSG signal is plotted alongside the Somnofy respiration curve for comparison, which is named PSG Respiration Thorax. As described in Section 2.3.1, a thoracic signal is a belt strapped around the chest which measures chest movements.

A plot of the Somnofy respiration curve, compared to a thoracic signal, is shown in Figure 5.7. Although not identical in terms of amplitude, the signals are correlated when it comes to frequency. Sleep-disordered breathing is indicative of the relative change in amplitude, given by the definitions of SDB scoring (Ruehland et al., 2009), which makes the respective difference in amplitude less important.

Plot 5.8 visualises an SDB event over the Somnofy respiration curve and PSG thorax signal. The SDB event plot is here artificially adjusted for readability. 0.5 corresponds to healthy, while the peak at 0.6 corresponds to an SDB event. A notable drop in respiration amplitude is observed right before the SDB event and increases right after the event finishes.

Figure 5.9 shows a larger section with a comparison between the Somnofy res-

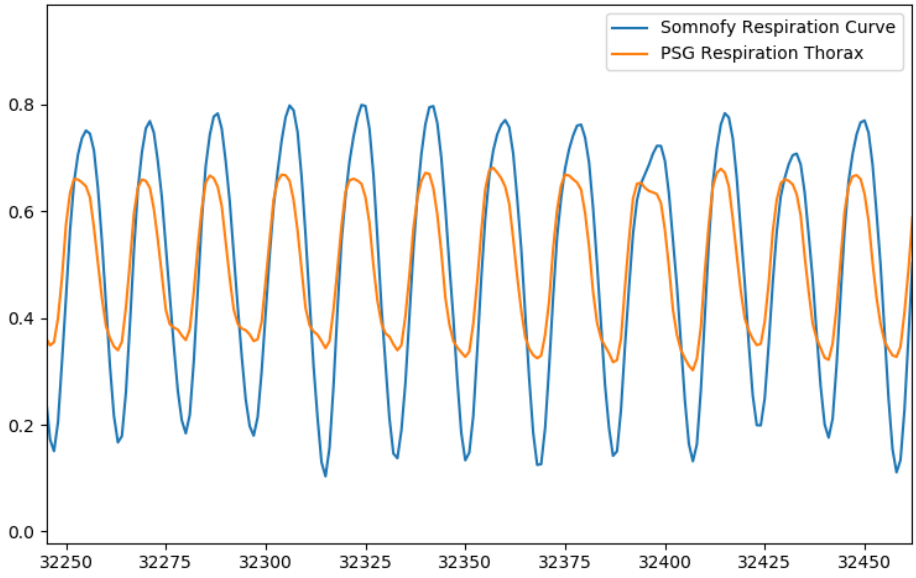


Figure 5.7: Plot of Somnofy respiration curve and a thorax belt

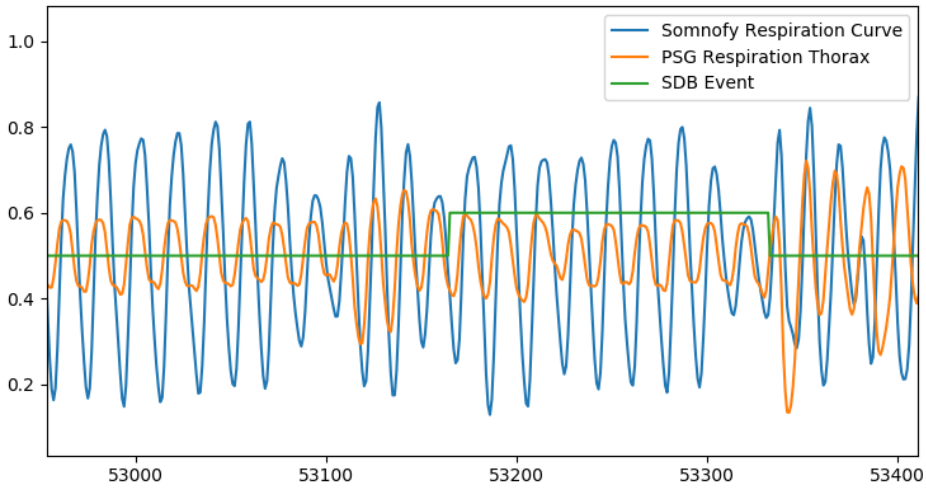


Figure 5.8: Comparison of the Somnofy respiration curve and a thorax belt signal during a SDB event

piration curve and thorax belt, where two events occur in the centre of the section. Clearly, there is a high correlation between the two respiratory signals. At the same time, several respiration spikes occur frequently throughout the data. Whether the spikes are SDB related and simply not substantial enough to be classified as an SDB event, or a naturally occurring phenomenon, is unknown. However, this shows that respiratory signal is somewhat irregular, or arrhythmic, without any event occurring. To only measure a decrease in respiration amplitude for the given signals are thus not likely as a sustainable approach for accurate recognition of SDB events.

Another feature available from the Somnofy trials is the physical movement of the patient. The movement signals describe any bodily movement of the subject. Four different movement signals are available. There are two base signals, which monitor movement within a time range of either 3 seconds or 20 seconds, respectively named *movement fast* and *movement slow*. As those signals are derived from Doppler-based radar signals, equivalent movements are recorded with different values depending on the distance to the radar. I.e. the same movement at a distance of 3 meters is recorded as smaller than the same movement at a distance of 1 meter. The normalization, which accounts for the relative distance,

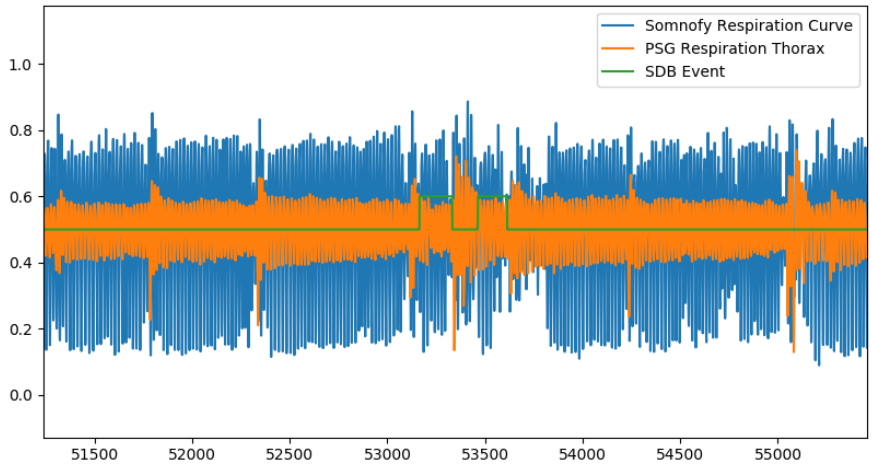


Figure 5.9: Large comparison of the Somnofy respiration curve and a thorax belt signal during two SDB events

is therefore applied to both base signals. Those signals are named *movement power fast* and *movement power slow*.

As the Somnofy respiration signal is computed using physical distances, a strong correlation between bodily movement and noise in the respiratory signal is expected. A close view of the respiratory signal alongside a Somnofy movement signal is given in Figure 5.10. The derivative of the movement signal is clearly greater than zero in the later parts of the graph, while the Somnofy respiration curve is not synchronised with the PSG thoracic signal. This shows the importance of monitoring bodily movements, as this highly affects the signal quality of the respiration data from Somnofy. Figure 5.11 shows a larger view of the same signals, but highlights in a bigger degree the correlation between bodily movements and noisy respiration signals.

Two other respiratory signals developed by Novelda are available, which are named *respiration rate* and *respiration distance*. These are quite accurate under given conditions, but has a substantial amount of missing data points where the signal is not available. This makes it troublesome to classify SDB events only relying on the Novelda respiration signals. As the signals are quite accurate for given periods, it might contribute to high-quality data for those given periods.

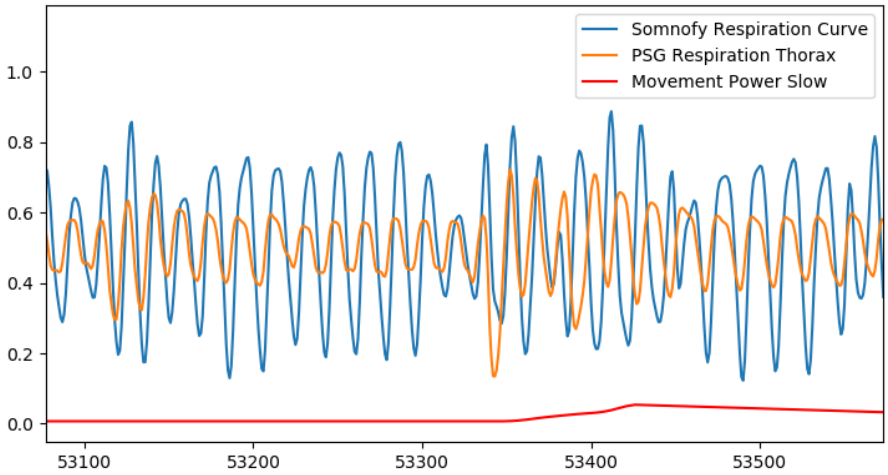


Figure 5.10: Plot of correlation between disturbances in respiration signal and bodily movement

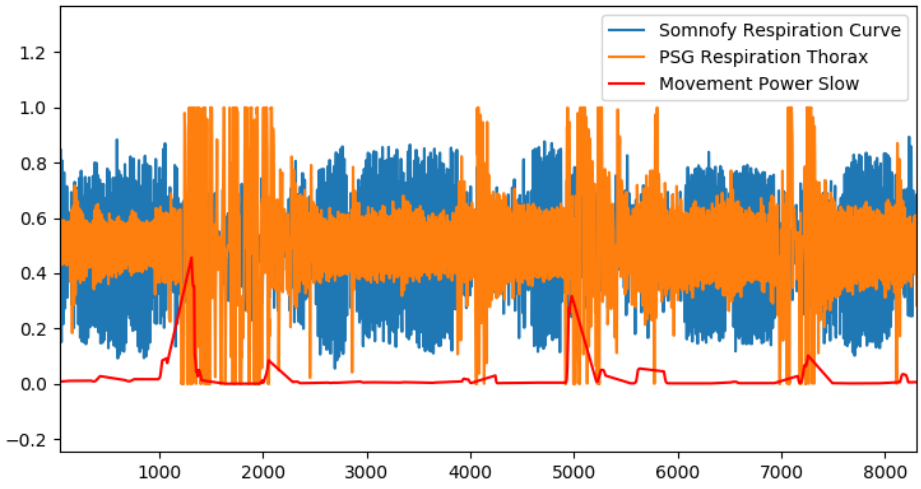


Figure 5.11: Large plot of correlation between disturbances in respiration signal and bodily movement

Another data feature used in the thesis is oxygen saturation. This signal is conventionally not available through radar but is easily integrated with the Somnofy system by using a wireless Bluetooth finger cap measuring oxygen saturation. The PSG oxygen saturation is therefore used as a temporary substitute, which often measures oxygen levels by the use of an equivalent sensor. Oxygen saturation is given as a percentage between 0-100. From the PSG trial, a second feature engineered oxygen saturation signal is available. The feature engineered oxygen saturation feature presents the change in oxygen during a saturation or desaturation period. As an example, if the oxygen level were to decrease from 97 to 93, the event signal would hold the value -4 throughout the period of desaturation.

Somnofy has a signal describing signal quality. Naturally, poor signal quality emphasis insecure readings, while a strong signal quality provides readings with high quality. While body movements might describe the confidence of the respiration data, signal quality describes the overall accuracy of any reading from Somnofy. Finally, a large range of environmental signals is also available for the Colosseum trial. The signals are self-describing, and are as follows:

- Temperature
- Pressure
- Humidity
- Sound Amplitude
- Ambient Light
- Red Light
- Green Light
- Blue Light

Chapter 6

Architecture and Model

The following chapter presents a detailed description of the model architecture used for detecting sleep-disordered breathing. The chapter is extended with a discussion of how the highly unbalanced data is processed for efficient training in Section 6.2. Finally, the post-processing applied to the predictions of the model is presented in Section 6.3.

6.1 The Model Architecture

Convolutional neural networks have been used in several successful papers on classifying sleep apnea from data sources similar to that of this thesis, as seen in Table 4.1. Most of them classify segments between 10 and 60 seconds, as either a healthy segment or a segment containing one specific SDB event. The achievement of these architectures is an indication of the strength of convolutional models for time series data in this field.

Based on the current research and best-performing methods available, a model based on convolutional neural networks was designed. The model as seen in Figure 6.1 consists of three different forms of blocks, two convolutional and one fully connected feedforward network. The first convolutional blocks of the model consist of three segments, the convolutional operation, batch normalisation and pooling. Each of these layers provides different functionality to the block. The

heart of the model is the convolutional layer, while batch normalisation and pooling have supporting roles in order to aid with training and generalisation. More specifically, batch normalisation both speeds up training time and improves generalisation by enabling a more stable learning scheme. Even though the actual reason for why batch normalisation is effective is currently the subject of debate (Santurkar et al., 2018), the technique is still shown to improve the performance of deep neural networks. The pooling in the first three blocks of the model is used for reducing the number of parameters in the model, which reduces the computational power needed for training. In addition, pooling does introduce some invariance to the translation of the features extracted by the convolutional layers in the early layers. When training the fully connected layers of the model, dropout is used in order to improve generalisation and prevent overfitting.

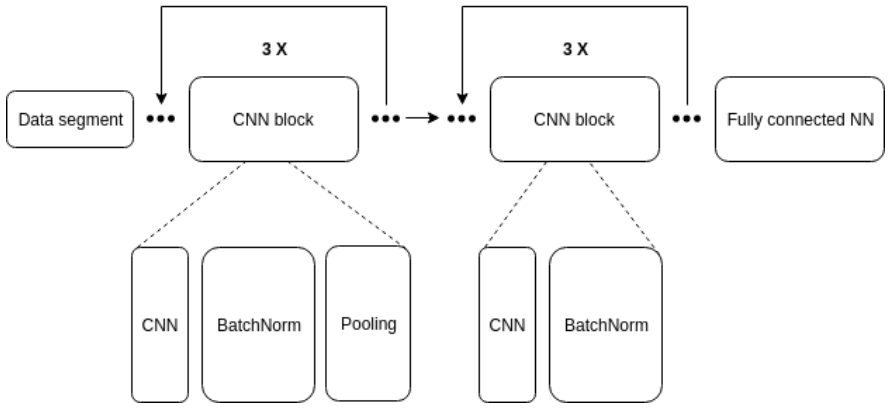


Figure 6.1: The general architecture of the CNN model. The model consists of three CNN-blocks with pooling, three CNN-blocks without pooling and a fully connected neural network.

When deciding for a model with a limited 30-second view of the data, some assumptions are made about the properties of the data with regards to the SDB events. The correct classification of an event requires the identification of two specific subpatterns, namely the beginning or the end of an event. If these subpatterns both occur outside of the local view of the model, the model might not identify an event correctly.

To see this, remember that the events are defined by the reduction of airflow compared to some mean value during the night. If the model only sees a subsection

of an event, there might not be a significant reduction present in the window as the flow of air is already lowered. Increasing the window size, and thus the size of the local view, so that all events are within one window is one way of mitigating this problem. This would increase the length of the time series the local model is given but results in a very long window, as some events can last for up to as long as two minutes. A window size of this magnitude would undoubtedly entail a large computational cost.

A solution that avoids an excessive window size, while incorporating the long term properties of the data, is to engineer additional features that span a longer period of time than the window. This allows the local model to detect a change from the current baseline in relevant features like breathing rate. Which long term features that are used in this thesis is presented in Section 6.2.2.

6.2 Data Processing

This section presents how the data is handled by the model. Section 6.2.1 presents how the training data is generated from the Somnofy data, while Section 6.2.2 presents the feature engineering. Section 6.2.3 discusses which techniques to apply for handling class imbalance.

6.2.1 Generating Training Data

The data available from both trials, as described in Section 5.1 and Section 5.2, comes in files generated by VitalThings. The files contain both the raw PSG and Somnofy readings, as well as the features created by Novelda and VitalThings. The data is read into data frames, reformated to the right time scale, and finally combined into one dataset.

Cleaning, Normalising and Scaling Values

After merging relevant features and signals, cells with missing data are interpolated linearly from the surrounding values where applicable, before any missing values are set to zero. The features signifying events, such as oximetry reduction, are scaled to be between 0 and 1 in value, while the remaining features are scaled by removing the mean and scaling to unit variance as in Equation 6.1. The scaled

value z is given by

$$z = \frac{x - u}{s}, \tag{6.1}$$

where x is a sample, u is the average of all samples, and s the empirical standard deviation of the samples.

Extracting events

Extracting events from the data is done in order to achieve a practical method for applying different balancing techniques. In order to build a collection of events, every segment where an event occurs is extracted and grouped by their class. When the event is extracted padding is added to both the beginning and end of the segment, in order to avoid losing data when applying a sliding window approach to extract same-size frames in the next stage of the processing pipeline.

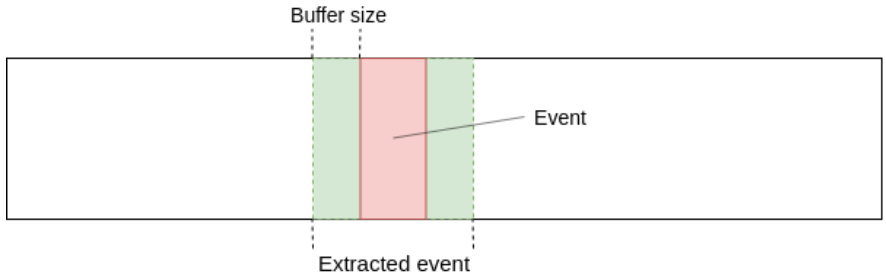


Figure 6.2: Events during a night is extracted with a buffer both pre- and post-event. Events are defined as a continuous section of data with corresponding labels of one of the apnea-classes. Buffer size is defined as half the window size used when extracting frames from extracted events.

As hypopnea and OSA events vary in size from approximately 10 to 90 seconds, each event is used to generate several frames containing 30 seconds of data. This is done in order to provide data with consistent size to the model. These sections, hereafter known as frames, are generated using a sliding window approach, with a stride of 3 seconds. With a larger working memory on the GPU, the stride could be reduced to generate a higher granularity of frames. Each frame is given a label, which corresponds to the ground truth of the timestep in the middle of the frame. The ground truth is given by the physicians scoring during the manual inspection of the PSG data. The events are then picked by a data sampler, that provides

data in batches for the model using one of the sampling methods discussed in Section 6.2.3. Healthy data is treated in the same way as any SDB event. Each segment of healthy data that is longer than a given threshold is extracted as an event, and sampled using the same technique as for the SDB events.

6.2.2 Feature Engineering

To aid the model in its effort recognising SDB events, a series of signals has been created using the Somnofy respiration curve. Neurokit (Makowski, 2016) is used for feature engineering and signal processing of the respiration curve. Neurokit is a library used for neurophysiological signal processing, abstracting many low-level signal processing techniques into a high-level toolbox for signal analysis of biosignals.

The Somnofy radar outputs two respiration signals of the subject being monitored. One originates from the radar chap itself, which is produced by Novelda. The other signal is generated from raw data by Somnofy. VitalThings state that the Somnofy respiration curve better correlates with the equivalent respiration curve from the PSG, and is chosen as the raw signal from which features are extracted. By using Neurokit, two valuable features are extracted; The first is a smoothed version of the original respiration curve, which removes any clear erroneous measurements. While the Neurokit library is built for preprocessing a raw respiratory signal directly from PSG and not the Somnofy respiration curve, a smoothing filter is still beneficial for improving overall signal quality. Throughout any given night, noise in the data is unavoidable. A large portion of this noise is due to bodily movements, which affect the signal quality greatly. In order to reduce the effects of noise sources such as movement, filters are applied.

The second feature extracted from Neurokit is the instantaneous respiration rate. It is reproduced at the exact same sampling rate as the respiration curve and describes the number of breaths per minute for that specific point in time. Respiration rate can fluctuate naturally throughout the night, and subjects might have different respiration rates due to natural variation. Therefore, a second respiration rate is computed manually, where the mean respiration rate in a 10-minute large window is determined. The model only has a 30-second long view of the data, and can not recall any long term dependencies, which is why these must be generated in advance. In addition, the baseline respiration rate is an important variable as the apnea-events are defined using the closely related variable of

air flow into the lungs. Creating features for respiration rate also simplifies the model's internal representation of the target function, as part of the target function is simplified by generating pre-computed patterns such as respiration rates. The model no longer has to learn how to map the respiration signal to the frequency of the respiration rate, which is an important feature used in diagnosing sleep apnea.

6.2.3 Data Imbalance Techniques

As presented in Section 4.3 only a few studies within SDB detection handled the data imbalance problem. The undersampling technique is seemingly more frequently used in SDB research, while other general systematic studies favour oversampling. There seems to exist a tendency for oversampling being the most effective (Buda et al., 2018), but systematic comparisons are only providing fairly vague general guidelines. On that regard, it seems only reasonable that the most suitable imbalance technique for this context is found through empirical results.

Basic undersampling and oversampling have shown promising results and straight forward to implement, as noted in Section 4.3.2. As the methods are fairly different in their approach, both imbalance techniques are included. A third hybrid approach is also suggested. The hybrid method oversamples all minority classes to the size of the largest minority, and undersamples the majority class from the original dataset randomly for each epoch. By using the proposed hybrid approach, one can retain all the information in the minority classes, in addition to achieving balanced training without a too large degree of duplication.

6.3 Post Processing of Output

As the granularity of the predictions for the proposed model is rather high compared to the existing methods in literature, a method for post-processing predictions was seen fit to implement. Using a few selected filters to both smooth and correct predictions, noise and irregularities in the predictions of the model might be reduced. As the definition of both obstructive sleep apnea and hypopnea require an event length of more than 10 seconds, a filter removing shorter events could decrease false positives, and therefore improving the precision of the predictions. Other filters such as a smoothing function to reduce irregular spikes or

dips in the predictions might also improve the performance of the model. After a set of filters are applied to the predictions, the AHI is calculated in order to provide a reasonable metric for whether the model is able to assess the severity correctly. The filters are summarised below.

Combining close predictions

Predictions for either obstructive sleep apnea or hypopnea, with fewer timesteps than a predetermined threshold between them, are combined. The filter is effectively smoothing the positive predictions for obstructive sleep apnea and hypopnea where predictions of either are interrupted by short segments of healthy predictions.

Filter out small predictions

Predictions with length less than the predetermined threshold are removed and replaced by healthy predictions. Using a threshold equivalent to 10 seconds removes all predictions shorter than the required length for it to be an event of obstructive sleep apnea or hypopnea.

Filter out too large predictions

Filter out events with a length larger than a predetermined threshold, and replace with healthy predictions. Sleep-disordered breathing is often contained within a 30-second interval. The longest events found for the data corpus used in this thesis is around 120 seconds. Excessively long events are improbable and therefore removed by the filter.

Chapter 7

Experiments and Results

This chapter presents a detailed view of the experiments conducted during this thesis, in light of the research questions and goals set out in Chapter 1.2. The experimental plan is given in Section 7.1, while the details about the setup of the experiments are presented in Section 7.2. Finally, the results are presented in Section 7.3.

7.1 Experimental Plan

The goal of the thesis is to predict sleep-disordered breathing using the Somnofy radar and achieving comparable results to related research using the gold standard PSG. Three research questions derived from the initial goal was originally proposed, and in order to answer the research questions, and to evaluate the capability of the model presented in Chapter 6, four experiments were designed and executed.

Research questions 1 Which features are the most informative in the detection of the different variations of sleep-disordered breathing?

In order to investigate Research Question 1, an experiment is designed utilising a random forest classifier to rank the importance of each available feature. The results are an important step in empirically validating the explicit domain knowledge that has been used to generate features from the available raw signals. Using

a combination of these results and explicit domain knowledge, a set of the most important features can be selected. This will allow both faster training times, as well as removing unwanted noise from irrelevant features.

Research question 2 What is the de facto standard method to minimise the effects of highly imbalanced datasets in multiclass classification?

After performing the feature importance ranking, the next experiment addresses Research Question 2. Using state of the art literature three techniques for balancing unbalanced data were chosen, and an experiment constructed in order to test the applicability and performance of each method on the dataset. The result of the experiment dictates which technique is utilised for the balancing of the dataset in the following experiments. By performing this experiment, valuable insight can be gained for several aspects of the architecture and data processing. The sensitivity of the model with regards to duplicates of the data, the importance of how much of the majority class data is included during training, and the feasibility of running computationally demanding data balancing schemes such as oversampling on the highly unbalanced dataset.

Research question 3 What are the current best performing approaches for classifying sleep-disordered breathing?

Building on the results of the previous experiment, and the literature study conducted when answering research question 3, various models are put under discussion leading to the proposed architecture in Section 6.1. This model is refined and evaluated through two experiments, the first being a random hyperparameter search to identify good model and training parameters, the second an evaluation of performance on the test dataset. The hyperparameter search enables the model to perform optimally, by empirically testing the performance of different combinations of hyperparameters. The evaluation experiment is used to evaluate the robustness and performance of the model on unseen data. Optimal training time is found by training the model with the train and validation set, after which the train and evaluation set are combined, and the model retrained on the combined set. Finally, the model is evaluated using the test set. The results of the validation-test are given both with and without post-processing of the model predictions, as described in Section 6.3. The results of the final evaluation experiment are evaluated and discussed further in Chapter 8.

7.2 Experimental Setup

The following section describes the setup for the experiments described in Section 7.1. The core libraries and tools used for executing the experiments are listed in Table 7.1, while the hardware and software used during the experiments are given in Table 7.2.

Library	**Version**
jupyter	1.0.0
neurokit	0.2.0
numpy	1.16.2
python	3.6.7
scikit-learn	0.20.2
scipy	1.2.1
seaborn	0.9.0
sklearn	0.0
torch	1.0.1

Table 7.1: Key libraries with their respective versions used to implement and execute the model and experiments

Entity	**Version**
OS	Ubuntu 18.04.2 LTS
CPU	Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
GPU	Tesla P100-PCIE-16GB
CUDA	10.0

Table 7.2: The hardware and software specifications of the computing platform on which the experiments are executed.

7.2.1 Data and Event Extraction

In order to create the dataset for training, validation, and testing, the two datasets described in Chapter 5, are combined and shuffled randomly. The training data is normalised and scaled as described in Section 6.2, and the scaling parameters from training and validation used when scaling the test data. Events are extracted with a 15-second buffer size, and frames generated with a window

size of 30 seconds. As creating all possible frames for every event would result in a very high number of frames, the frames for training events are generated with a sliding window approach with a 3-second step length. The frames from events used for testing are generated with a one-second step length so that the predictions can be made with a one-second granularity. Every frame is grouped with frames of the same class, which allows training examples to easily be sampled using the different sampling techniques as described later in this chapter.

7.2.2 Feature Importance Study

To estimate the importance of each of the features used by the model, a feature importance study using random forests is conducted. The machine learning library Sklearn (Pedregosa et al., 2011) provides the method and implementation of the random forest classifier and feature ranking. The Random forest uses the default number of one hundred estimators, and the automatically balanced weights created by the Sklearn implementation to assign each class with a weight inversely proportional to the class frequency of the training data. A seed is provided for the random generator in order to obtain consistency when testing. The remaining parameters are not assigned and thus given their default value provided by the implementation. The parameters used are shown in Table 7.3.

Parameter	**Value**
Estimators	100
Weights	”Balanced”
Random Seed	3

Table 7.3: Parameters used with the Sklearn implementation of RandomForestClassifier when performing the feature importance experiment.

7.2.3 Model Architecture

The details of the model used for the data sampling experiment is presented in Table 7.4. The initial parameters were found to achieve stable results during initial training and testing of the model. Adam is used as the optimisation function, while the loss is computed using Cross Entropy Loss. Both the optimiser

and the loss function is provided by pytorch and uses the default parameters unless specified in Table 7.5.

7.2.4 Data Sampling Techniques

The data sampling experiment empirically validates which imbalance technique is the most efficient. The three techniques tested are random *oversampling*, random *undersampling* and *hybrid sampling*. Only the most relevant features are included, based upon the results from the feature importance experiment. The search is conducted with a 5-fold cross-validation, using early stopping with a patience of 50 epochs. To test the performance on a specific fold, the model of that fold with the lowest validation loss during training is used. The parameters used for training are listen in Table 7.5.

7.2.5 Random Hyperparameter Search

The random hyperparameter search will be conducted with the parameters given in Table 7.6. The search is done with the sampling technique showing the best results following the experiments on data imbalance techniques.

The hyperparameter search was run for a total of 36 rounds, where each round a combination of parameters was sampled without replacement from the set of all possible combinations. Hence, 36 distinct sets of hyperparameters were tested and evaluated.

7.2.6 Model Evaluation

In order to evaluate the performance of the model, the model is trained and evaluated with the data imbalance method and parameters found in the previous experiments. The original dataset was split up into a training set, validation set, and test set. Initially, the test set was created by randomly extracting 15% of the data. Then, 15% of the remaining data was randomly extracted to create the test set. The model is trained and evaluated using the train and validation set, finding the optimal number of epochs for training. The train and validation set is then combined, the model retrained using the found number of epochs, and the model tested using the test set.

Layer	Input	Output	Kernel	Stride	Padding	Activation	Param
Conv	12*150	30*150	3	1	1	-	-
Pooling	30*150	30*151	2	1	1	-	-
BN	30*151	30*151	-	-	-	Relu	-
Conv	30*151	30*151	3	1	1	-	-
BN	30*151	30*151	-	-	-	Relu	-
Pooling	30*151	30*152	2	1	1	-	-
Conv	30*152	30*152	3	1	1	-	-
BN	30*152	30*152	-	-	-	Relu	-
Pooling	30*152	30*153	2	1	1	-	-
Conv	30*153	30*154	2	1	1	-	-
BN	30*154	30*154	-	-	-	Relu	-
Conv	30*154	30*155	2	1	1	-	-
BN	30*155	30*155	-	-	-	Relu	-
Conv	30*155	30*156	2	1	1	-	-
BN	30*156	30*156	-	-	-	Relu	-
Reshape	30*156	4680	-	-	-	-	-
Dropout	4680	4680	-	-	-	-	- p=0.2
Linear	4680	256	-	-	-	Sigmoid	-
Dropout	256	256	-	-	-	-	- p=0.2
Linear	256	128	-	-	-	Sigmoid	-
Linear	128	64	-	-	-	Sigmoid	-
Linear	64	3	-	-	-	Softmax	-

Table 7.4: Layers and parameters of the convolutional neural network model. The model consists of batch normalisation layers (BN), one dimensional convolutional layers (Conv), Pooling layers (Pool), Dropout layers, Linear layers and a reshape layer that flattens the data when transforming the output on the convolutions. All in- and output shapes are without the Batch dimension of the data, and is given as *features* $*$ *length*.

Parameter	Value
Multistep Learning rate milestones	30, 80, 130
Multistep learning rate gamma	0.1
Learning rate	0.005
Batch size	128

Table 7.5: Parameters and their values used when performing the 5-fold cross validation experiment for evaluating data sampling techniques. Learning rate milestones describes at which epoch the learning rate is multiplied with the learning rate gamma in order to reduce the learning rate.

Parameter	Range / Values	Stepsize	Nr choices
Dropout	0.1 - 0.5	0.1	5
Batchsize	32 - 512	Base-2 log scale	5
Learning rate Milestones	[10,20,30], [15,30,50],[20,40,60]	-	5
CNN channels	7, 15, 20, 30, 50	-	5
SUM	-	-	375

Table 7.6: Parameters used in the random gridsearch. Learning rate milestones provides a list of at which epochs during training to reduce learning rate, while CNN channels yields the number of filters to use for the convolutional neural network.

Filter	Threshold in seconds
Combining close predictions	3
Filter out smaller than	10
Filter out larger than	120

Table 7.7: Threshold parameters used for post-processing filters.

7.2.7 Post-processing of Predictions

Post processing the predictions are done with the filters and parameters listed in Table 7.7. The filters are used on the predictions of the final model used for testing, in order to further improve its performance. They are applied in the sequential order in which they are listed.

7.3 Experimental Results

This subchapter presents the results of the experiments described in Section 7.1. The first four sections correspond to each of the aforementioned experiments, while Section 7.3.5 presents the results after post-processing the predictions of the model evaluation experiment.

7.3.1 Feature Importance

By performing the feature importance selection experiment described in Section 7.2, the following results are obtained as seen in Figure 7.1. The description of

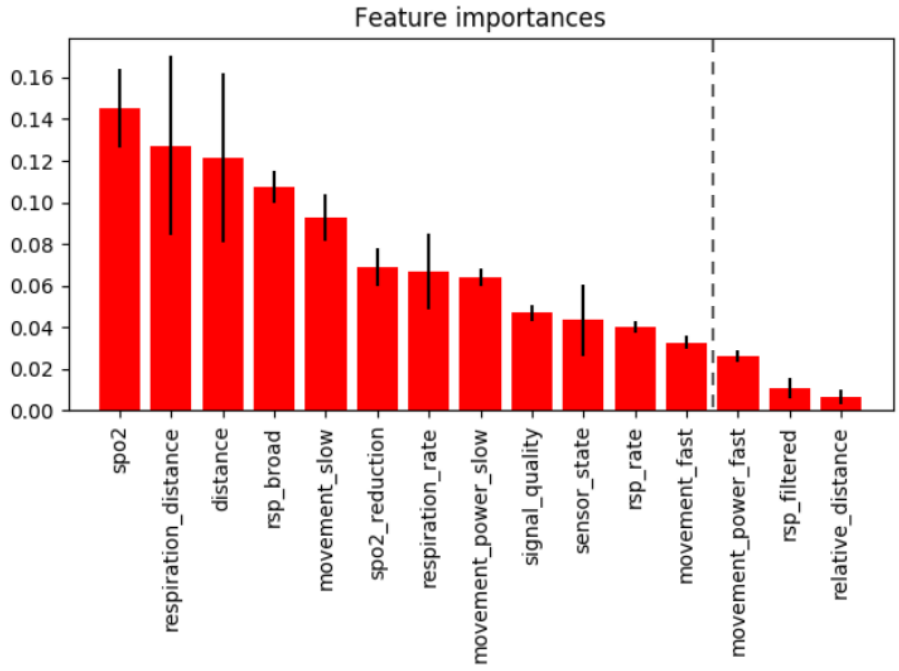


Figure 7.1: Plot of the feature importances. The black line of each feature describes the inter-tree variability, or the standard deviation between the importance of each feature in each classifier-tree in the forest. See Table 7.8 for a description of attribute names.

each feature is given in Table 7.8

The features are ranked from most to least to most important. As shown in the plot, the set of the most important features to take note of is the oxygen saturation, the Novelda respiration curve, distance to the patient, a wide sliding window mean of respiration rate and movement of the patient. Using these results, *relative_distance*, *rsp_filtered* and *movement_power_fast* were removed from the training data. *relative_distance* and *rsp_filtered* are the same signal, but where the latter is processed by Neurokit. Both signals describe respiration curve, which is already covered by *respiration_distance*, making them abundant due to high correlation. *movement_power_fast* is removed due to its close correlation to *movement_fast* which also has a low score with regards to importance.

Signal	Description
spo2	Oxygen saturation
respiration_distance	Novelda respiration curve
Distance	Physical distance to patient
rsp_broad	10-minute mean of respiration rate
movement_slow	Slow-paced monitoring of bodily movements
spo2_reduction	Change in oxygen saturation
respiration_rate	Novelda respiration rate
movement_power_slow	Normalised slow-paced monitoring of bodily movements
signal_quality	Signal quality for the Somnofy
sensor_state	State of the Somnofy sensor
rsp_rate	Neurokit respiration rate
movement_fast	Fast-paced monitoring of bodily movements
movement_power_fast	Normalised fast-pace monitoring of bodily movements
rsp_filtered	Neurokit filtered respiration curve derived from Somnofy respiration curve
relative_distance	Somnofy respiration curve

Table 7.8: Description attributes for Figure 7.1.

7.3.2 Data Sampling Techniques

Table 7.9 and 7.10 present the results from the data sampling experiment.

Sampling Technique	Class	Precision	Recall	F1-score
Undersampling	Healthy	0.99 ± 0.00	0.87 ± 0.03	0.93 ± 0.02
	Hypopnea	0.11 ± 0.03	0.48 ± 0.16	0.18 ± 0.06
	OSA	0.02 ± 0.02	0.38 ± 0.26	0.03 ± 0.03
Oversampling	Healthy	0.99 ± 0.00	0.82 ± 0.18	0.89 ± 0.13
	Hypopnea	0.11 ± 0.08	0.54 ± 0.10	0.17 ± 0.10
	OSA	0.01 ± 0.03	0.01 ± 0.06	0.02 ± 0.04
Hybrid	Healthy	0.99 ± 0.00	0.87 ± 0.07	0.92 ± 0.04
	Hypopnea	0.10 ± 0.04	0.57 ± 0.07	0.16 ± 0.05
	OSA	0.02 ± 0.02	0.25 ± 0.17	0.04 ± 0.04

Table 7.9: Precision, recall and F1-score for each class, using either undersampling, oversampling or a hybrid sampling approach. The experiment was conducted using a 5-fold cross-validation. The average value across all five folds is given for each metric, along with the standard deviation between the scores for each fold.

Sampling Technique	Accuracy	Cohen's kappa
Undersampling	0.86 ± 0.03	0.13 ± 0.05
Oversampling	0.82 ± 0.18	0.15 ± 0.11
Hybrid	0.86 ± 0.07	0.13 ± 0.05

Table 7.10: Accuracy and Cohen's kappa for each of the tested sampling methods. The experiment was conducted using a 5-fold cross validation. The average value across all five fold is given for each metric, along with the standard deviation between the scores for each fold.

The best performing imbalance technique based on empirical results is our suggested hybrid sampling approach, which is used in the subsequent experiments. As the choice between sampling techniques is non-trivial, the choice is discussed further in Section 8.1.

7.3.3 Random Hyperparameter Search Results

The random hyperparameter search ran for a total of 36 randomly picked combinations of hyperparameters. The top performers are listed in Table 7.11, and provides the information on which the choice of hyperparameters for the final model evaluation was based upon.

Milestones	Dropout	Batchsize	Channels	Class	Precision	Recall
20,40,60	0.2	32	7	Healthy	0.99	0.93
				Hypopnea	0.16	0.69
				OSA	0.00	0.00
15,30,50	0.5	64	15	Healthy	0.99	0.94
				Hypopnea	0.16	0.54
				OSA	0.09	0.35
20,40,60	0.3	128	30	Healthy	0.99	0.94
				Hypopnea	0.17	0.52
				OSA	0.11	0.41
10,20,30	0.5	256	50	Healthy	0.99	0.94
				Hypopnea	0.16	0.64
				OSA	0.01	0.01
10,20,30	0.5	512	15	Healthy	0.99	0.86
				Hypopnea	0.10	0.69
				OSA	0.00	0.02

Table 7.11: The best performing choices of parameters for the random hyperparameter search. Performance was compared based on the scores for precision and recall, yielding the most promising combinations of parameters for the final evaluation of the model. The best performing set of parameters is seen in the row marked in gray.

As the grid search is random and the set of parameters tested relatively limited, it is not expected that it will find the absolute Pareto-optimal set of parameters. Training time per session is also in terms of hours, making it difficult to set aside time and resources for a more comprehensive search.

7.3.4 Model Evaluation

When training on the train-validation datasets, the optimal training length was found to be 70 epochs. After combining the train and validation set to obtain the

new training set, the final model is trained and evaluated with the test dataset. The confusion matrix for the test data is shown in Figure 7.2.

The validation metrics for the model on the test data is shown in Table 7.12.

Class	Precision	Recall	f1-score	Accuracy	Cohen's kappa
Healthy	0.99	0.92	0.95		
Hypopnea	0.15	0.51	0.23	0.90	0.23
OSA	0.10	0.42	0.16		

Table 7.12: Performance metrics for the test set. Accuracy and Cohen's kappa are calculated for all classes in the full test set.

The AHI for each night is calculated, and the predictions are compared with the true AHI for each night in the test data. The results are seen in Table 7.13.

7.3.5 Model Evaluation after Post-processing

After post-processing, the AHI and confusion matrix for the test set is recalculated, and presented in Figure 7.3 and Table 7.15 respectively.

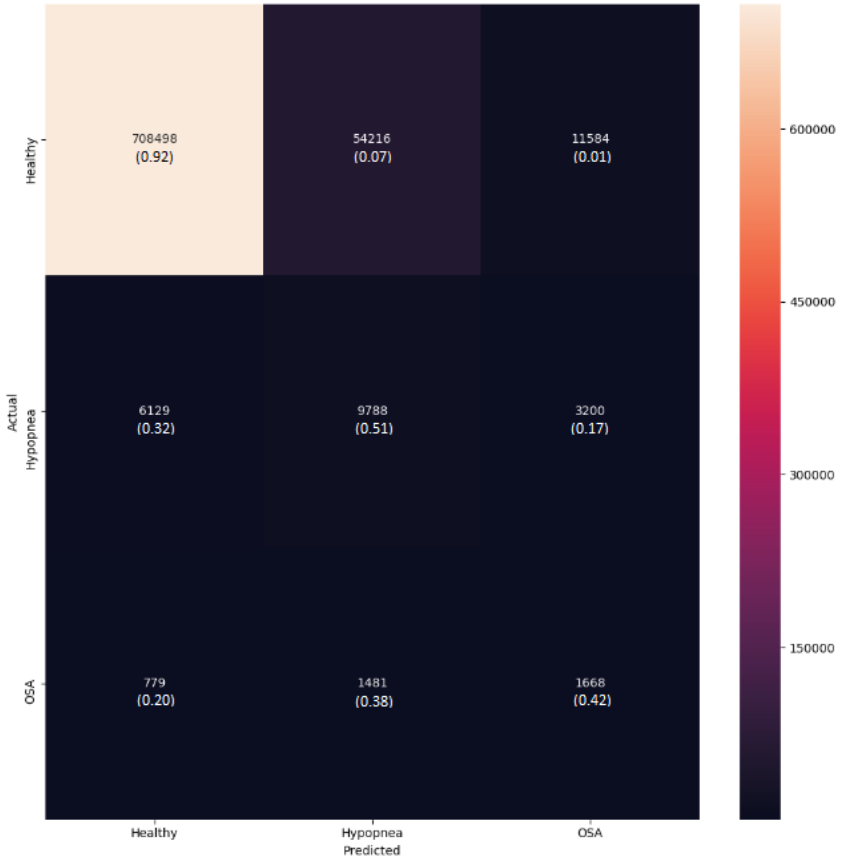


Figure 7.2: Confusion matrix for the test data on the final evaluation of the model. The normalised values for each row are shown in parentheses below each count.

Index	Predicted AHI	Labelled AHI	Error
Night 1	1.54	0.66	0.88
Night 2	2.83	0.77	2.06
Night 3	8.36	0.00	8.36
Night 4	9.17	0.00	9.17
Night 5	3.76	3.96	0.20
Night 6	10.43	0.00	10.43
Night 7	2.89	0.00	2.89
Night 8	5.53	2.45	3.09
Night 9	144.39	36.47	107.93
Night 10	4.05	0.00	4.05
Night 11	6.84	0.34	6.50
Night 12	4.59	5.38	0.80
night 13	2.55	0.49	2.07
Night 14	8.57	1.99	6.57
Night 15	21.19	13.42	7.77
Night 16	6.02	0.00	6.02
Night 17	1.32	0.00	1.32
Night 18	3.34	0.00	3.34
Night 19	6.70	0.00	6.70
Night 20	6.08	1.99	4.08
Night 21	6.57	0.00	6.57
Night 22	7.61	0.85	6.76
Night 23	2.19	0.49	1.70
Night 24	16.06	3.92	12.14
Night 25	21.56	14.04	7.52
Night 26	5.28	3.40	1.89
sum	-	-	230.81
std	26.93	7.54	20.06
mean	12.29	3.49	8.88

Table 7.13: AHI for each night in the test-data. The mean and standard deviation is calculated from each metric, as well as the sum of errors of all nights.

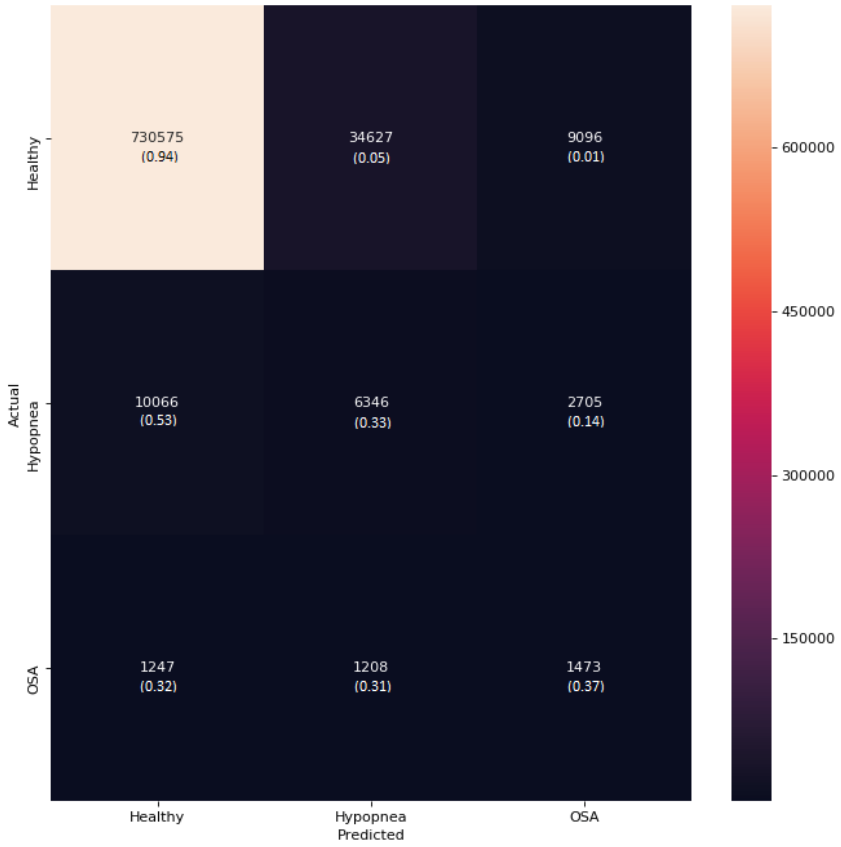


Figure 7.3: Confusion matrix for the test data on the final model, after post processing. The normalised values for each row are shown in parentheses below each count.

Class	Precision	Recall	f1-score	Accuracy	Cohen's kappa
Healthy	0.98	0.94	0.96		
Hypopnea	0.15	0.33	0.21	0.93	0.22
OSA	0.11	0.38	0.17		

Table 7.14: Performance metrics for the validation set after post-processing. Accuracy and Cohen's kappa are calculated for all classes in the full test set.

Index	Predicted AHI	Labelled AHI	Error
Night 1	1.32	0.66	0.66
Night 2	1.55	0.77	0.77
Night 3	4.85	0.00	4.85
Night 4	4.90	0.00	4.90
Night 5	2.23	3.96	1.73
Night 6	5.68	0.00	5.68
Night 7	1.16	0.00	1.16
Night 8	3.83	2.45	1.38
Night 9	61.08	36.47	24.61
Night 10	2.58	0.00	2.58
Night 11	4.26	0.34	3.92
Night 12	2.59	5.38	2.79
night 13	1.09	0.49	0.61
Night 14	5.78	1.99	3.79
Night 15	10.54	13.42	2.87
Night 16	3.15	0.00	3.15
Night 17	0.88	0.00	0.88
Night 18	1.96	0.00	1.96
Night 19	3.60	0.00	3.60
Night 20	4.68	1.99	2.69
Night 21	3.36	0.00	3.36
Night 22	3.59	0.85	2.75
Night 23	1.34	0.49	0.85
Night 24	8.22	3.92	4.30
Night 25	9.86	14.04	4.18
Night 26	4.02	3.40	0.63
sum	-	-	90.65
std	11.28	7.54	4.47
mean	6.08	3.49	3.49

Table 7.15: AHI for each night in the test-data after post-processing of predictions. The mean and standard deviation is calculated from each metric, as well as the sum of errors of all nights.

Chapter 8

Evaluation and Conclusion

This chapter contains the evaluation of the results obtained and the conclusion of the thesis. The results are evaluated in Section 8.1, and discussed in Section 8.2. In addition, the contributions of the thesis are listed in Section 8.3, and the recommendations for future work are presented in Section 8.4

8.1 Evaluation

In order to evaluate the results in Section 7.3, each experiment and its results are presented in order of execution. The result of each experiment is evaluated in light of the research question it was designed to answer. The results are discussed further in Section 8.2.

8.1.1 Data Imbalance Techniques

The results of the imbalance experiment are shown in Table 7.9. Due to the limited size of the dataset, a choice was made to perform a 5-fold cross-validation, instead of the generally adopted 10-fold cross-validation. Some folds had few cases of the minority classes in the training data which increased the risk of overfitting. Despite this, the experiment still yielded valuable results.

Undersampling and hybrid sampling performed relatively similar, however hybrid sampling has a far better recall for hypopnea. In comparison, undersampling had significantly lower recall with a substantially higher standard deviation. High recall is a critical performance metric, due to the detrimental effect of false negatives when using the system as a screening tool. Therefore, hybrid sampling is favoured as it provides more stable results, and has a better recall with a non-substantial reduction in precision.

The results indicate clearly that oversampling performed the worst. A detailed inspection of the folds shows that oversampling had a clear tendency to learn one class only while discarding the others, which can be explained by the narrow distribution of apnea and hypopnea events. Another problematic aspect of oversampling is the increased need for computational resources. 5-fold cross validation with oversampling used approximately 3 days to finish, while the hybrid and undersampling approach finished in a matter of hours. Currently, this makes oversampling the least favourable approach for dealing with the class imbalance.

In general terms, the imbalance techniques do not contribute to a better fundamental understanding of how to detect hypopnea and OSA. The imbalance approach is rather an optimisation technique and is not directly aiding the model in getting a fundamental understanding of the SDB classes. It is also important to note that the imbalance ratio for undersampling was not optimised.

8.1.2 Features

Research Question 1 addressed which features that are of the greatest importance when detecting sleep apnea. As certain features are highly correlated, such as the described respiration features, some of them might be deemed unnecessary. This implies that the signal itself might be highly relevant for recognising sleep-disordered breathing, but is less informative than other correlated signals when compared to the remaining features. As seen in Figure 7.1, oxygen saturation, respiration curve, physical distance, a wide mean of respiration rate and movement are the most relevant features.

Oxygen saturation and respiration amplitude are part of the key definition of hypopnea and apnea. It is therefore expected that both oxygen saturation and respiration curve are among the most relevant features. Bodily movements are highly correlated with noisy readings in the respiration data, which makes movement another expected feature. It is somewhat unexpected that the 10-minute

view of the respiration rate is ranked as one of the most important features. One hypothesis to why this feature is important is that the 10-minute baseline of respiration rate correlates strongly with the other variables, such as airflow, which is the most significant variable used when diagnosing apnea-events. Physical distance is in itself, to the best of the authors' knowledge, irrelevant for SDB events, but is related to the signal quality and reading vital signs, as Somnofy is a radar-based system. Physical distance affects every raw and extracted signal, which provides a natural explanation for why it is ranked as an important feature.

8.1.3 Hyperparameter Search

Using the Hybrid sampling method, a grid search with sets of randomly picked parameters was conducted. The results are shown in Table 7.11 and shows some of the top performing selection of parameters. The best set of hyperparameters is seen in the row marked with Gray, which achieved the best balance between precision and recall for both hypopnea and OSA. While other sets of parameters performed better at hypopnea, such as with an increased number of channels and increased batch size, these combinations often lead to the model discarding the minority class OSA. This is seen in both set 1, 4 and 5 in Table 7.11. It is also worth mentioning that the model now is considerably more precise than during the sampling technique experiment.

8.1.4 Results

After the grid search was executed, the best parameters were chosen and the final model trained and validated. The confusion matrix for the validation predictions are presented in Figure 7.2, the validation metrics are presented in Table 7.12, and the AHI prediction results are presented in Table 7.13.

As the confusion matrix for the test data show, the model achieves a relatively low precision when predicting events of disease at a one-second granularity. One property of the confusion matrix is that the model seems to perform poorly at separating healthy data and hypopnea. In addition, the model seems to perform worse at separating healthy and hypopnea than it does for separating healthy and OSA. There is, however, a clear overweight of hypopnea compared to OSA in the dataset, so one might expect the model to perform better at classifying hypopnea. One argument for why the model performs better at predicting OSA

is that OSA entails a larger and more sudden change on a subjects behaviour and physiological measurements, and is, therefore, more easily separable from the healthy data than hypopnea.

As expected there is a clear trade-off between precision and recall, where in order to achieve a good enough recall, the precision is reduced. This is a well-known dilemma in the field of machine learning, and one that was expected to occur due to the nature of the domain. Looking at the performance metrics in Table 7.12, the indications observed from the confusion matrix is confirmed. As expected the model performs well on the majority class, while it performs worse on the minority classes. It is also worth noting that the model achieves better recall and precision on hypopnea than on OSA, which is discussed at a later point in this chapter. Overall accuracy is high, due to the models' ability to be accurate on classifying events from the majority class, while Cohen's kappa is low due to the models lacking the ability to separate between the different classes.

Plotting the predictions and the corresponding labelled apnea-events reveals both the strength and weakness of the model. As seen in Figure 8.1 the model is aware of the periods when events occur, but as the reader might notice the precision regarding exactly when the events occur is not perfect.

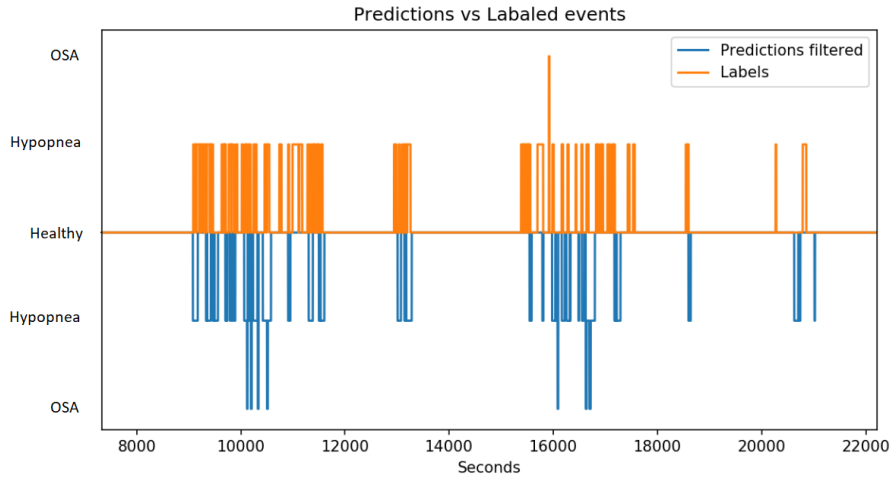


Figure 8.1: Plot of the Hypopnea/OSA predictions and labels for a 4-hour segment of night 15 of the test set.

When looking closer at the predictions, as seen in Figure 8.2, the model is also unable to differentiate between a large event and a sequence of adjacent short events. In addition, a false prediction of an event takes place after the series of hypopnea events end. This might be an area with symptoms of apnea, which is not serious enough to qualify as an event in an of itself, but where the model identifies the indications or symptoms.

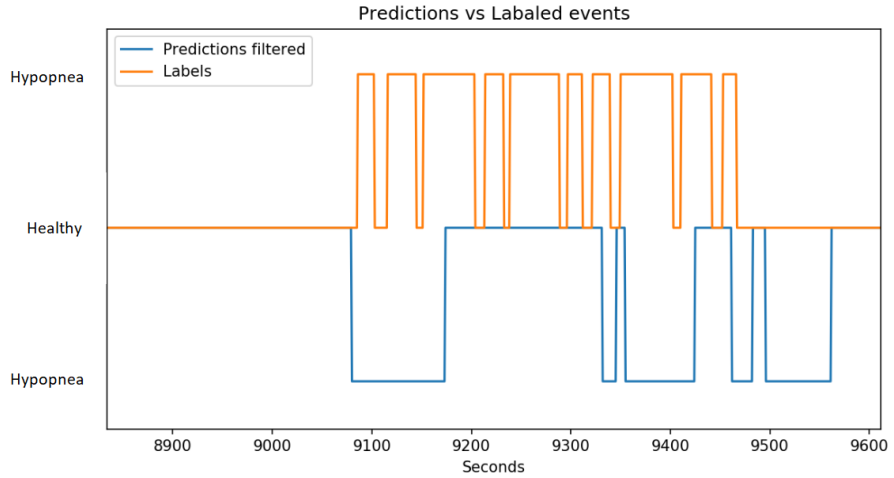


Figure 8.2: Plot of the Hypopnea/OSA predictions and labels for a 10-minute segment of night 15 of the test set.

Looking at the AHI-scores for the test-set in Table 7.13, the model consistently predicts a higher AHI-score than the actual score. For nights with no events, such as nights 3 and 4, it predicts a high AHI-score resulting in a large error. For the three nights with the largest AHI, namely nights 9, 15 and 25, the model predicts high AHI scores. This suggests that it is, in fact, possible to extract useful predictions of the AHI score from the model which operates on a one-second granularity. The mean and standard deviation is quite large, considering that each class of the AHI-severity index is separated by 5 units. However, note that night 9 skews that distribution, with a significantly high error.

As the combination of predictions with a 1-second granularity using a non-PSG data signal is previously untested in the field, there are no results available to which the performance of the proposed model can be compared directly. There

are some earlier studies that are comparable to the approach taken, which will provide some insight into its performance. A common practice in these studies is to calculate the sensitivity and specificity on the degree of severity of a patient's AHI, which will be the only metric directly comparable to the results in this thesis.

PSG Based Systems

When comparing to PSG based systems there are several potential candidates for comparison. Van Steenkiste et al. (2018a) developed an LSTM model using various numbers of PSG signals in order to extract respiratory features. The system used 30-second epochs with a 1-second stride but does not report the performance on this granularity. Instead, the study reports results on AHI severity, with recall for severe, moderate, mild and normal AHI-class being 73%, 54%, 34%, and 3% respectively. Precision for the respective severities were 62%, 49%, 51%, and 13%. Despite the few numbers of mild, moderate and severe AHI cases seen for the test set in Table 7.15, it is still possible to claim that the AHI-severity results are comparable. A larger validation set with a better distribution of AHI-severity would provide better empirical evidence, but that requires a specific selection of patients in order to balance the test set.

Urtnasan et al. (2018b) provided predictions on 10-second epochs, with a multi-class classification of both Hypopnea and OSA. The paper achieved better precision and recall for both Hypopnea and OSA, but lower precision and recall for the healthy class. Considering the source of the data being ECG and the relatively large segment size compared to the 1-second granularity of this thesis, the results seem promising.

A different approach was taken by Dey et al. (2018), which used a convolutional neural network to perform a binary prediction on healthy or apnea, for 1-minute segments. Using 1-channel ECG, the study achieved a recall of 97.8% and a true negative rate of 99.2 %. The study does not provide AHI-severity scores. With high performance on predictions of one-minute granularity, there is a high probability of accurate AHI-severity predictions as well. Comparing the results of this thesis to that of Dey et al. (2018) is not straight forward, but some conclusions can be drawn. In general, the proposed model does not perform as good as the top performers among PSG-based systems. However, compared to most PSG-based systems, the approach taken in this thesis adds complexity by using radar-based systems, a much finer granularity, and by training the classifier to separate between hypopnea and OSA.

Non-PSG Based Systems

The comparison is now shifted towards non-PSG systems, as this allows the results to be compared to methods with similar sources of data. As expected, non-PSG based systems perform poorly compared to PSG-based systems. The most comparable radar system, SleepMinder (Zaffaroni et al., 2009), achieves a recall of 46.4%, 25.3%, 66.7% and 62.5% for severe, moderate, mild and normal AHI-severity respectively. Compared to these results, it is reasonable to claim that the implemented system could perform comparably or better than Zaffaroni et al. (2009) when it comes to predicting AHI-severity.

Post-Processing

Performance metrics using post-processing is presented in Table 7.14, and shows noticeably effect on recall for hypopnea. The overall accuracy is increased by 3% suggesting that a large number of wrong positive predictions for hypopnea was filtered out. Despite the reduction in recall for the non-healthy classes, the AHI predictions are improved significantly. The mean and standard deviation of the errors between predicted and actual AHI-score is reduced, while the predictions for nights with high AHI scores are still relatively high compared to the nights with a low AHI. For a visualisation of the effect of post-processing see Figure 8.3.

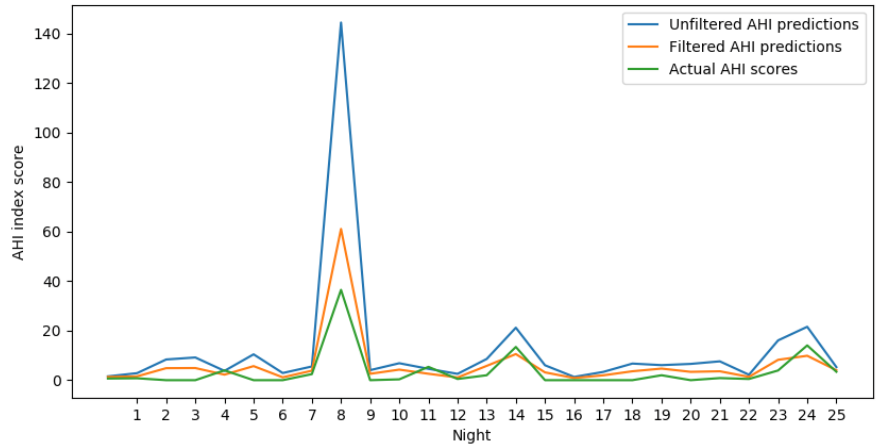


Figure 8.3: AHI predictions for each night for both post-processed and unprocessed model predictions.

8.2 Discussion

This chapter discusses the previously evaluated results and the performance of the model architecture. The feature importance rankings are discussed in Section 8.2.1, the data and data imbalance in Section 8.2.2. The architecture and composition of the model is discussed in Section 8.2.3, while its predictions and results after post-filtering are discussed in Section 8.2.4

8.2.1 Feature Importance

Part of the results from the feature importance experiment was unexpected. The Somnofy respiration curve and respiration rate were expected to be one of the most relevant features, due to SDB being closely related to respiration. However, they were the least important features deemed by the random forest. Novelda's respiration distance was ranked as the 2nd most valuable feature. Referring to dialogue with VitalThings, Novelda's respiration curve was described as quite accurate, but with a notable degree of missing signals, as discussed in Section 5.3. It is therefore suggested that the Novelda respiration signal is more accurate than the Somnofy respiration signal and that any missing signals tend to occur in noisy periods where there is a small chance for any SDB events. However, all three features describe the same respiration curve. It is somewhat natural for a decision tree, or random forest, to only devise one of the features as important, following the discussion of information gain in Section 3.6. That would explain why the other two respiration signals are ranked poorly. This can be empirically tested by discarding the Novelda respiration distance signal, and check whether one of the other respiratory signals achieves a higher ranking. This would imply that respiration data is highly beneficial in the process of recognising SDB events.

Oxygen saturation is graded as the most relevant feature in SDB recognition. Saturation of oxygen is, alongside respiration volume, the key definition of apnea and hypopnea. It is therefore expected that oxygen saturation is valued highly, due to its tight coupling with SDB events by nature. Contrary to these expectations, the feature engineered signal describing changes in saturation (*spo2_reduction*) is ranked as relatively unimportant. *spo2_reduction* is a feature engineered signal from the raw oxygen saturation mentioned above, and contains less information than the raw signal, which calls for a natural explanation for why *spo2_reduction* is less preferred. It is also important to note that the two signals are correlated,

implying that when one becomes relevant, the other becomes less significant, following the discussion of information gain and decision trees. This is can also be tested empirically in later research, discarding the *spo2* signal to see whether *spo2_reduction* gains higher importance relative to the other signals. In conclusion, this shows that oxygen saturation is another key metric when working with SDB events.

Distance is ranked as the 3rd most valuable signal. In a conventional setting, the distance between a radar and a person is irrelevant for the occurrence of SDB events. However, noise in radar measurements has a high correlation with body movements, which is effectively captured by the distance feature. Presumably, distance becomes an important feature as it describes when sudden changes in especially respiration rate, are related to imprecise measurements or an actual change in respiration. Distance is therefore presumed as relevant due to the use of radar when monitoring the patient.

Respiration rate computed by a mean 10-minute sliding window is ranked as the fourth most relevant signal. The sliding window computes an average baseline respiration rate for the current moment. Respiration rate is tightly coupled with sleep stages, referring to Section 2.1.1. It is therefore reasonable that a broader view on respiration rate, compared to the momentous respiration rate ranked as the 11th most important feature, is more important. Namely, because respiration rate varies naturally throughout the night, and any deviations in respiration rate must be interpreted from the current baseline. In conclusion, the respiration baseline is a valuable metric when recognising SDB events.

Finally, *movement slow* is ranked as the 5th most important feature. As for the *distance* feature, movement is tightly correlated with noise when monitoring vital signs. It is therefore unlikely that those movement measurements would be just as relevant for PSG trials, which applies an intrusive monitoring technique. Referring to Figure 7.1, the other movement features are ranked as quite irrelevant. This is likely a result of high correlation with the other signals. It is however interesting to see that the *power* signals are ranked as the least relevant signal for both *slow* and *fast* paced readings. This can be a result of applying scaling to all movement signals, which then minimises the effect of distance when reading the degree of movement, as discussed in Section 5.3. *Distance* is also already ranked as highly relevant, which is, in fact, a measure of distance, meaning the algorithm might have figured out that distance and movement are related. In conclusion, movement becomes an important signal when recognising SDB events when us-

ing radar, namely due to the correlation between bodily movements and noisy signals.

Signal quality and sensor state are ranked as moderately relevant. Poor signal quality and a nontrivial sensor state do naturally affect all signals recorded. On the other hand, this shows that no feature is single-handily able to recognise SDB events. When the signal quality and sensor state has a relative importance of around 40% of the most important signal, it is implied that the data are hardly separable and that a joint set of features must be combined for effective recognition.

8.2.2 Data Imbalance

Oversampling was initially thought of as a beneficial imbalance technique for the given data corpus. A review of sampling techniques claimed that oversampling is expected to perform the best, except for anomaly detection problems with extremely high imbalance (Buda et al., 2018). The results of the current experiments do however show that hybrid and undersampling performs on par, and are definitely more accurate than oversampling.

As previously discussed, oversampling tended to overfit, which explains why it fails to generalise hypopnea and OSA events. That is rather unexpected, as CNN was suggested as resistant to overfitting when applying oversampling (Buda et al., 2018). A synthesis for why oversampling performed the worst, is that the imbalance ratio is too great or that the distribution of SDB events is too poor.

Another sampling technique considered was SMOTE. However, the achievements or results in the literature was deemed not significant enough compared to that of the less complex methods and imposes a significant cost with regards to implementation. As a result of that, SMOTE was not included. For the same reasons, using generative adversarial networks (GANs) were not considered, as the cost of implementation does not seem to cover up for the minor gain in accuracy (Suh et al., 2019). Modifying the classifier itself, e.g. through cost-based learning, is less observed in the literature. However, Buda et al. (2018) claimed that neither of those techniques would outperform oversampling, except for anomaly detection problems with extremely high imbalance. With that in mind, it was favourable to focus on the simpler techniques which took less time to implement, before evaluating any of the later steps. As down sampling majority classes are found as the

most beneficial approach, it is currently far more relevant to consider techniques as SMOTE where undersampling is a key step.

There are reasons to believe that picking samples by a heuristic is more efficient than random undersampling. This would ensure that the events sampled covers the domain and problem space the most efficiently. As roughly 96% of the data consists of healthy periods, it would seem beneficial to sample segments evenly throughout the problem space. An easy example is the fluctuations in respiration rate which correlates with sleep stages. As the respiration rate is an effective feature for recognising SDB events, ensuring even sampling of healthy data from all sleep stages could be beneficial for the model's domain representation.

8.2.3 Model Architecture

There are a number of challenges that arise when attempting to classify events with temporal dependencies, using a model with a limited local view. However, as seen in Figure 8.2, it is clear that despite the local properties of the model it is not able to distinguish gaps between the events and events themselves. One cause of this could be that the local window of 30 seconds is providing too much information to the model in cases with multiple transitions between events and healthy segments. This complicates the task at hand for the model, namely to identify subpatterns or transitions where an event start or end. Furthermore, in order to incorporate long term dependencies into the model, the mean breathing rate for a sliding window of 10 minutes was generated. This feature represents the pre-event baseline of the breathing rate but does not capture any broader temporal dependencies such as the average breathing rate for the entire night. If the baseline was reduced after a period of subsequent events, the model might not detect new events due to the low ratio between the current baseline and the new event. Finally, there might simply not be enough data available for learning complex situations, where a number of events take place in a short duration of time.

In order to learn the temporal dependencies without using feature engineering or other bias-prone approaches, an attempt was done at implementing a combined CNN-LSTM model, which would capture both local and temporal dependencies. A convolutional model similar to that of in Chapter 6.1 was trained, the convolutional layers of the model frozen, and the output of the model used as the input to a recurrent model using LSTM-cells. Using sequential frames from whole

nights as input to the combined model, the LSTM was trained. Unfortunately, given the overweight of healthy data when training with whole nights of data, the combined model was not able to avoid what is similar to overfitting on the healthy majority class. Inevitably, the CNN-LSTM model yielded nothing but healthy predictions.

Recurrent neural network models, specifically the LSTM variant, have been successfully used for rare event detection in time series data (Malhotra et al., 2015). However, these results are obtained on data on the scale of hundreds of seconds, not thousands, as is the case in this thesis. Most other literature using recurrent neural networks for detecting sleep apnea such as Cheng et al. (2017) does so not on the raw signals, but the features extracted from it, which provides a much smaller time-scale, or fewer steps to learn, for the recurrent network. Another version is training the recurrent network on 30-second epochs such as done in Van Steenkiste et al. (2018a), but this model is only able to model the intra-epoch temporal dependencies.

8.2.4 Predictions and Post-filtering

One night of sleep for a subject with severe sleep apnea can consist of a highly complex series of events. As seen in Figure 8.4, some patients with severe apnea can have over 30 events an hour. In these situations, it can be a highly complex task to distinguish between events and the small pauses between them, since there is a high degree of fluctuation in features such as breathing rate and bodily movement.

This leads to the two main problems of the model for these types of situations. Either it does not have the representational power required to model the complex behaviour of high frequent apnea events, or it does not have enough data for training. The same difficulty of predicting correctly in the period between events is seen in Figure 8.5, where the model predicts a segment of disease that has a duration of over 250 seconds. This is obviously a too long continuous segment of events, but the model is unaware of any past, present or future predictions and hence unable to adjust its output accordingly.

Using the visual inspection of plotted predictions, as seen in Figure 8.2, it seems evident that the model is unable to precisely detect when an event starts or stops in sections with frequent states of disordered breathing. Training more selectively

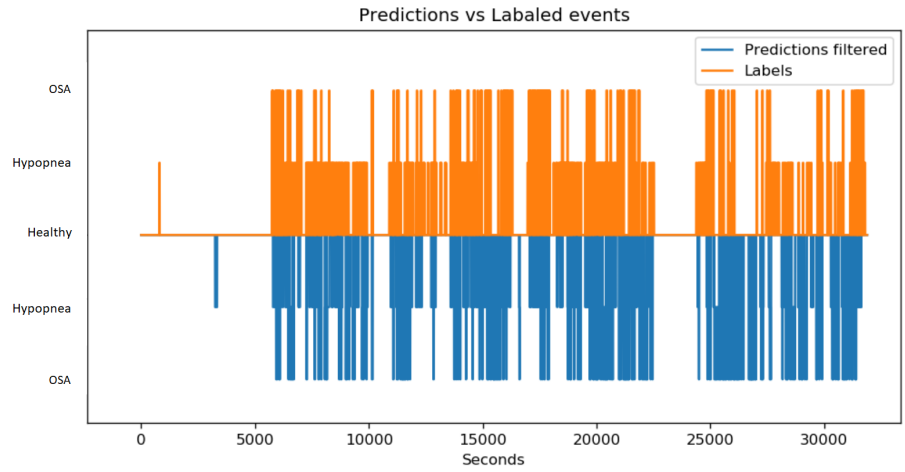


Figure 8.4: Visual comparison of OSA/Hypopnea predictions and the diagnosed events for night 9 in the test data. This subject has severe sleep apnea and in average more than 18 events per hour.

on these transitions between events and healthy segments will most likely improve the models' ability to separate the healthy majority class from the diseases.

Filtering

As with many other techniques, the post-processing of predictions has both benefits and drawbacks. It is shown in Section 7.3 that AHI predictions are dramatically improved by filtering. As seen in Figure 8.6 filtering also improves the performance on certain segments dramatically, compared to the unfiltered version in Figure 8.5.

There are several pitfalls of the relatively naive approach taken when applying filtering. One example of this is shown in Figure 8.7, where filtering removes the prediction entirely due to its length being over a given threshold.

Many of these pitfalls are hard to predict before encountering them visually and provide an extra layer of complexity to the architecture. If the model was able to perform the equivalent of post-processing as a part of the models' internal representation of the target function, the use of post-processing could be avoided. There are several ways of combining logical rules with deep neural networks, one

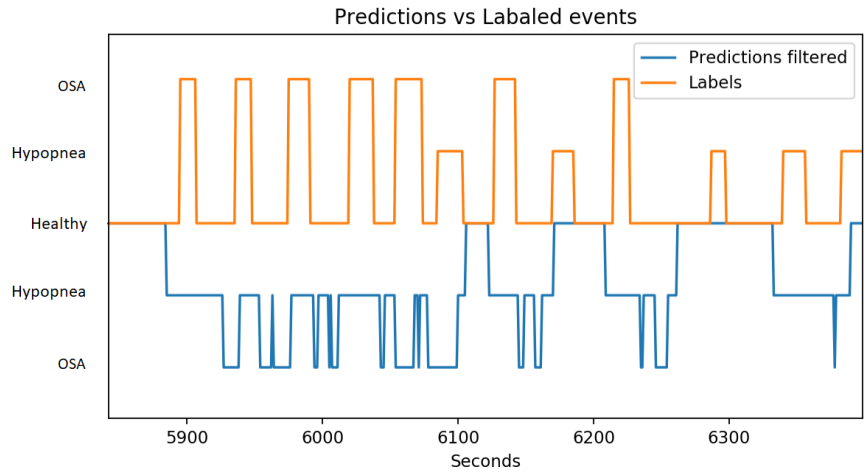


Figure 8.5: Visual comparison of OSA/Hypopnea predictions and the diagnosed events for night 9 in the test data. The model wrongly predicts a continuous segment of apnea-events lasting over 250 seconds.

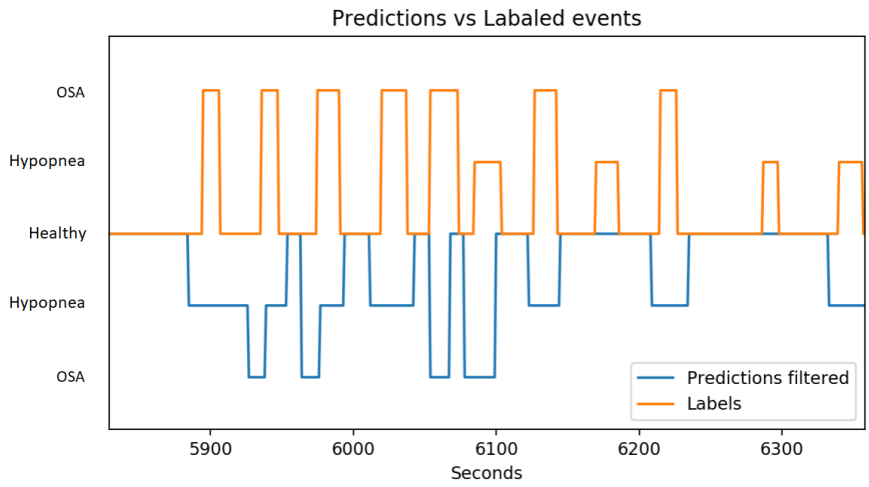


Figure 8.6: Predictions after applying post-processing compared to the diagnosed events, for night 9 in the test data. Filtering removes the excessively long continuous sequence of apnea-events seen in Figure 8.5.

of which incorporates logical rules in a student-teacher architecture by enforcing the student to emulate the predictions of the rule-regularised teacher (Hu et al., 2016).

8.3 Contributions

Throughout the duration of this thesis, the authors wanted to push not only the current methods forward, but to open up a new area in the field of detecting sleep-related disorders. By moving from detecting apnea in long segments of time, to attempting to detect apnea at a much finer granularity, the diagnostic power of radar-based sleep analysis applications such as Somnofy is improved. This expansion of the field, combined with the future improvements discussed in Section 8.4, are the most significant contributions of this thesis.

Although the model created does not perform well enough to be considered for clinical screening, it represents a building block upon which other research can be built. While, to the authors' knowledge, no other research on the topic has

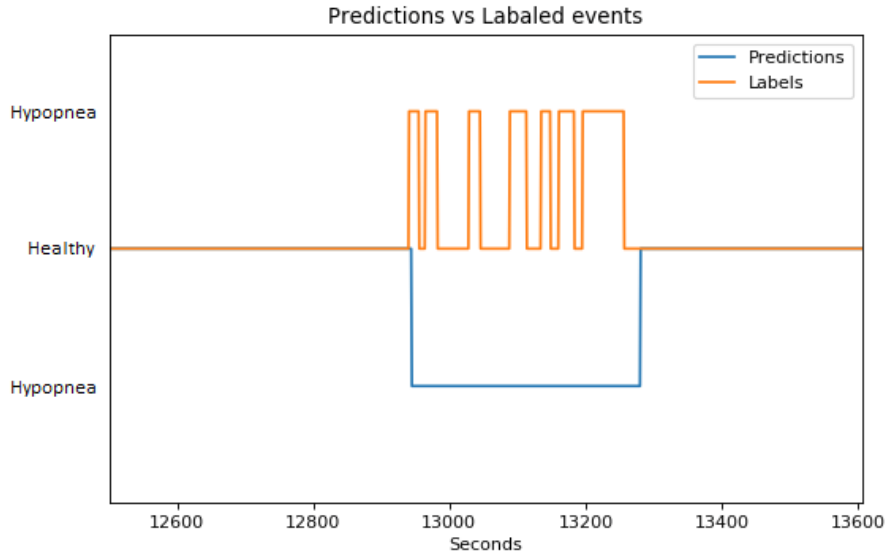


Figure 8.7: Visual comparison of the predictions and the diagnosed events for night 15 in the test data. The post-processing removed the entire prediction as its length is bigger than the given threshold of 160 seconds.

provided a thorough analysis of data imbalance regarding SDB recognition, this thesis provides a complete solution. Through the conducted feature importance study, a series of different features were ranked. Respiration curves, oxygen saturation, physical distance, respiration rates, and bodily movements were ranked as the most important features when using Doppler-based radar for SDB analysis. The findings are only of minor importance for research using PSG-based methods, except for the confirmation that respiration and oxygen saturation are important features. However, the findings are important for radar-based methods as signal quality and noise, represented by physical distance and movement signals, are highly important features in SDB recognition.

The challenges faced during the implementation of the combined convolutional and recurrent model also leaves behind an unanswered question, which might encourage future researchers to find solutions. Using a joint model is theoretically justified but without convincing results from the present study. It is also interesting to see how the broad respiration rate is deemed important by the feature importance study, as it represents one method for incorporating long term memory into a local model. The thesis further supports the use of convolutional neural networks on non-Markovian time-series data within this domain. Feature engineering, and with it capturing long-term temporal dependencies, is still an important contribution to the success of local models.

8.4 Future Work

This chapter contains the suggestions for future work for the model and architecture presented in Chapter 6. While most of the future work became apparent during the implementation of the model and execution of the experiments, some post-experiment improvements were also uncovered during the writing of the thesis.

8.4.1 Data, Imbalance and Sampling

Two different data sources are used jointly in the thesis. Natural variations might exist in the signals due to different variables such as the radars position and orientation in the room, or environmental or seasonal differences such as temperature, light, and sounds. In addition, the radar can either be mounted to the wall or put

on a level surface near the bed, which emphasizes the importance of an orientation independent model. Mixing two different datasets with a different distribution of disease and population, where both the data from and operating of the device might differ consistently, does provide the model with information on which dataset a subject originates from. By using this information, the model can learn that a subject from one dataset is more likely to belong to a group with a high degree of disease, and hence predict disease with a greater probability. Somnofy as an HSAT would also be operated very differently by each individual user, as they do not follow the same procedural rules as one would in a clinical setting. Zhao et al. (2017) provided a solution for this problem when using an adversarial training architecture in order to remove information that would allow the model to distinguish one source from another. Using an encoder-discriminator technique the encoder was trained to remove all information using a discriminator to predict the origin of the data as the feedback to the encoder. This same strategy could be used in order to remove both intra-dataset variations but also inter-dataset variations such as the previously mentioned environmental and operational variables.

A different but related concern throughout the duration of this thesis has been the data imbalance. The number of recorded nights is low, and the events within them sparse. Acquiring new data is not an option, as this is both time-consuming and costly, as is the case within many other fields of medical research. It is, however, possible to generate data synthetically, such as proposed in SMOTE (Chawla et al., 2002). This would contribute to a more uniformly distributed dataset, and generate more training examples for the model. Based on the results of both over and undersampling in the conducted experiments, the dataset contains too few cases of especially OSA for these methods to successfully work. SMOTE might mitigate some of these problems related to what is most likely overfitting on a set of few cases for one particular class.

Another very important aspect of this highly imbalanced dataset is that only a few per cent of the data are highly relevant for identifying the subpatterns in the data related to SDB events. While most of the healthy data consist of long periods of healthy breathing, which is relatively simple to classify, it is the few per cent of healthy data that lie between and around events that are difficult to classify. Training on more of these cases specifically would make the model more proficient in classifying edge cases correctly. One way to do this is to choose what healthy data to include in the test data with certain constraints, or weighting the probability of sampling edge cases higher than for healthy data from long periods

with no events.

8.4.2 Feature Engineering

One of the strengths and driving causes behind the great success of neural networks is the ability to extract features, patterns, and correlations from raw data without explicitly engineering features suitable for the given domain. Doing so runs the clear risk of discarding information valuable to the model, and hence reducing its predictive power. As the suggested model here is a convolutional neural network, it is especially well fitted for taking in raw signals with minimal engineering. However, neural networks are data-driven, which means that the model requires a given amount of data before it is able to learn the domain. Feature engineering can be used in order to aid the model to comprehend the domain, by reducing the noise to signal ratio of the data. It is debatable whether the data of the corpus given in this thesis is sufficient enough for recognizing SDB events at the given granularity, or whether the domain is complex to the degree that feature engineering is required for feasible training times and computational requirements. Possible improvements reside in the results of investigating whether better feature engineering is beneficial for the model.

As the proposed model has a limited local view of the data, features were made to incorporate both local features and features representing a wider view. One example is the 10-minute baseline of respiration rate, which would provide a baseline upon which to compare the current respiration rate in the local view. As this feature was proven valuable in the feature importance experiment, more emphasis can go into creating additional features to provide a more global view on the data, as well as engineering features for other signals.

8.4.3 Hyperparameter Search

Random grid search is in no way an optimal optimization algorithm for model-hyperparameters. Despite a random search being more effective than a sequential grid-search (Bergstra and Bengio, 2012), there are still more effective choices available. Bayesian Optimization (Snoek et al., 2012) is one clear candidate for hyperparameter optimization that could result in a gain in performance due to more optimized hyperparameters.

Bibliography

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. (2016). Deep speech 2: End-to-end speech recognition in English and Mandarin. In *International conference on machine learning*, pages 173–182.

Avcı, C. and Akbaş, A. (2015). Sleep apnea classification based on respiration signals by using ensemble methods. *Bio-Medical materials and engineering*, 26(s1):S1703–S1710.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(1):281–305.

Bianchi, M., Lipoma, T., Darling, C., Alameddine, Y., and Westover, M. (2014). Automated sleep apnea quantification based on respiratory movement. *International journal of medical sciences*, 11(8):796.

Bixler, E. O., Vgontzas, A. N., Lin, H.-M., Ten Have, T., Rein, J., Vela-Bueno, A., and Kales, A. (2001). Prevalence of sleep-disordered breathing in women: effects of gender. *American journal of respiratory and critical care medicine*, 163(3):608–613.

Buda, M., Maki, A., and Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Cheng, M., Sori, W. J., Jiang, F., Khan, A., and Liu, S. (2017). Recurrent neural network based classification of ECG signal features for obstruction of sleep apnea detection. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, volume 2, pages 199–202. IEEE.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Conrad, I., S, A.-I., and S, Q. (2007). The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications. *American Academy of Sleep Medicine*.

Davidovich, M. L. Y., Karasik, R., Tal, A., and Shinar, Z. (2016). Sleep apnea screening with a contact-free under-the-mattress sensor. In *2016 Computing in Cardiology Conference (CinC)*, pages 849–852. IEEE.

Dey, D., Chaudhuri, S., and Munshi, S. (2018). Obstructive sleep apnoea detection using convolutional neural network based deep learning framework. *Biomedical engineering letters*, 8(1):95–100.

Domingos, P. (1999). Metacost: A general method for making classifiers cost-sensitive. In *KDD*, volume 99, pages 155–164.

Eckert, D. J., Jordan, A. S., Merchia, P., and Malhotra, A. (2007). Central sleep apnea: pathophysiology and treatment. *Chest*, 131(2):595–607.

Erman, M. K., Stewart, D., Einhorn, D., Gordon, N., and Casal, E. (2007). Validation of the ApneaLink^{TM} for the screening of sleep apnea: a novel and simple single-channel recording device. *Journal of Clinical Sleep Medicine*, 3(04):387–392.

Franklin, K. A. and Lindberg, E. (2015). Obstructive sleep apnea is a common disorder in the population—a review on the epidemiology of sleep apnea. *Journal of thoracic disease*, 7(8):1311.

Fu, R., Zhang, Z., and Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328. IEEE.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Gutiérrez-Tobal, G. C., Álvarez, D., Crespo, A., Del Campo, F., and Hornero, R. (2019). Evaluation of machine-learning approaches to estimate sleep apnea severity from at-home oximetry recordings. *IEEE journal of biomedical and health informatics*, 23(2):882–892.

Haidar, R., Koprinska, I., and Jeffries, B. (2017). Sleep apnea event detection from nasal airflow using convolutional neural networks. In *International Conference on Neural Information Processing*, pages 819–827. Springer.

Hassan, A. R. and Haque, M. A. (2017). An expert system for automated identification of obstructive sleep apnea from single-lead ECG using random under sampling boosting. *Neurocomputing*, 235:122–130.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2015). Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Hillman, D. R., Murphy, A. S., Antic, R., and Pezzullo, L. (2006). The economic cost of sleep disorders. *Sleep*, 29(3):299–305.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8).

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hu, Z., Ma, X., Liu, Z., Hovy, E., and Xing, E. (2016). Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.

Hudson Sleep & TMJ Center (2019). Obstructive Sleep Apnea. http://sleeptmjdr.com/wp-content/uploads/2013/06/Airway-Illustration-Crop.jpeg. [Online; accessed 13-march-2019].

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Isied, A. and Tamimi, H. (2015). Using random forest (rf) as a transfer learning classifier for detecting error-related potential (errp) within the context of p300-speller. In *Bernstein Conference*.

Janbakhshi, P. and Shamsollahi, M. (2018). Sleep apnea detection from single-lead ECG using features based on ECG-derived respiration (EDR) signals. *IRBM*, 39(3):206–218.

Javaheri, S., Smith, J., and Chung, E. (2009). The prevalence and natural history of complex sleep apnea. *Journal of Clinical Sleep Medicine*, 5(03):205–211.

Kagawa, M., Ueki, K., Tojima, H., and Matsui, T. (2013). Noncontact screening system with two microwave radars for the diagnosis of sleep apnea-hypopnea syndrome. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2052–2055. IEEE.

Koley, B. L. and Dey, D. (2013a). Automatic detection of sleep apnea and hypopnea events from single channel measurement of respiration signal employing ensemble binary SVM classifiers. *Measurement*, 46(7):2082–2092.

Koley, B. L. and Dey, D. (2013b). Real-time adaptive apnea and hypopnea event detection methodology for portable sleep apnea monitoring devices. *IEEE Transactions on Biomedical Engineering*, 60(12):3354–3363.

Kotsiantis, S., Kanellopoulos, D., Pintelas, P., et al. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36.

LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer.

Levendowski, D., Steward, D., Woodson, B. T., Olmstead, R., Popovic, D., and Westbrook, P. (2009). The impact of obstructive sleep apnea variability measured in-lab versus in-home on sample size calculations. *International archives of medicine*, 2(1):2.

Li, J., Gao, L., Song, W., Wei, L., and Shi, Y. (2018a). Short term traffic flow prediction based on LSTM. In *2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 251–255. IEEE.

Li, K., Pan, W., Li, Y., Jiang, Q., and Liu, G. (2018b). A method to detect sleep apnea based on deep neural network and hidden markov model using single-lead ECG signal. *Neurocomputing*, 294:94–101.

Makowski, D. (2016). *NeuroKit: A Python Toolbox for Statistics and Neurophysiological Signal Processing (EEG, EDA, ECG, EMG...)*.

Malhotra, P., Vig, L., Shroff, G., and Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. In *Proceedings*, page 89. Presses universitaires de Louvain.

Martín-González, S., Navarro-Mesa, J. L., Juliá-Serdá, G., Kraemer, J. F., Wessel, N., and Ravelo-García, A. G. (2017). Heart rate variability feature selection in the presence of sleep apnea: An expert system for the characterization and detection of the disorder. *Computers in biology and medicine*, 91:47–58.

Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., and Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks*, 21(2-3):427–436.

Novelda (2015). *X2 Impulse radar transceiver*. Novelda, Trondheim, Norway.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pigou, L., Van Den Oord, A., Dieleman, S., Van Herreweghe, M., and Dambre, J. (2018). Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*, 126(2-4):430–439.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Rechtschaffen, A. and Kales, A. (1968). A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. *National Institutes of Health (U.S) 204*.

Resta, O., Foschino-Barbaro, M., Legari, G., Talamo, S., Bonfitto, P., Palumbo, A., Minenna, A., Giorgino, R., and De Pergola, G. (2001). Sleep-related breathing disorders, loud snoring and excessive daytime sleepiness in obese subjects. *International journal of obesity*, 25(5):669.

Reza, M. S. and Ma, J. (2018). Imbalanced Histopathological Breast Cancer Image Classification with Convolutional Neural Network. In *2018 14th IEEE International Conference on Signal Processing (ICSP)*, pages 619–624. IEEE.

Richard, M. D. and Lippmann, R. P. (1991). Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483.

Rosen, I. M., Kirsch, D. B., Chervin, R. D., Carden, K. A., Ramar, K., Aurora, R. N., Kristo, D. A., Malhotra, R. K., Martin, J. L., Olson, E. J., et al. (2017). Clinical use of a home sleep apnea test: an American Academy of Sleep Medicine position statement. *Journal of Clinical Sleep Medicine*, 13(10):1205–1207.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Ruehland, W. R., Rochford, P. D., O'donoghue, F. J., Pierce, R. J., Singh, P., and Thornton, A. T. (2009). The new AASM criteria for scoring hypopneas: impact on the apnea hypopnea index. *Sleep*, 32(2):150–157.

Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Sadek, I., Seet, E., Biswas, J., Abdulrazak, B., and Mokhtari, M. (2018). Non-intrusive vital signs monitoring for sleep apnea patients: A preliminary study. *IEEE Access*, 6:2506–2514.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 2483–2493. Curran Associates, Inc.

Savage, H. O., Khushaba, R. N., Zaffaroni, A., Colefax, M., Farrugia, S., Schindhelm, K., Teschler, H., Weinreich, G., Grueger, H., Neddermann, M., et al. (2016). Development and validation of a novel non-contact monitor of nocturnal respiration for identifying sleep-disordered breathing in patients with heart failure. *ESC heart failure*, 3(3):212–219.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., and Napolitano, A. (2010). RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197.

Sharma, H. and Sharma, K. (2016). An algorithm for sleep apnea detection from single-lead ECG using Hermite basis functions. *Computers in biology and medicine*, 77:116–124.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc.

SSB, editor (2013). *Statistisk årbok 2013*. Statistisk sentralbyrå.

Stone, W. C. (1997). Electromagnetic signal attenuation in construction materials. Technical report, National Institute of Standards and Technology.

Strine, T. W. and Chapman, D. P. (2005). Associations of frequent sleep insufficiency with health-related quality of life and health behaviors. *Sleep medicine*, 6(1):23–27.

Suh, S., Lee, H., Jo, J., Lukowicz, P., and Lee, Y. O. (2019). Generative Oversampling Method for Imbalanced Data on Bearing Fault Detection and Diagnosis. *Applied Sciences*, 9(4):746.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Toderici, G., Vincent, D., Johnston, N., Jin Hwang, S., Minnen, D., Shor, J., and Covell, M. (2017). Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5306–5314.

Tran, V., Al-Jumaily, A., and Islam, S. (2019). Doppler radar-based non-contact health monitoring for obstructive sleep apnea diagnosis: A comprehensive review. *Big Data and Cognitive Computing*, 3(1):3.

Tripathy, R. (2018). Application of intrinsic band function technique for automated detection of sleep apnea using HRV and EDR signals. *Biocybernetics and Biomedical Engineering*, 38(1):136–144.

Uddin, M., Chow, C., and Su, S. (2018). Classification methods to detect sleep apnea in adults based on respiratory and oximetry signals: a systematic review. *Physiological measurement*, 39(3):03TR01.

Urtnasan, E., Park, J.-U., and Lee, K.-J. (2018a). Automatic detection of sleep-disordered breathing events using recurrent neural networks from an electrocardiogram signal. *Neural Computing and Applications*, pages 1–10.

Urtnasan, E., Park, J.-U., and Lee, K.-J. (2018b). Multiclass classification of obstructive sleep apnea/hypopnea based on a convolutional neural network from a single-lead electrocardiogram. *Physiological Measurement*, 39(6):065003.

Van Steenkiste, T., Groenendaal, W., Deschrijver, D., and Dhaene, T. (2018a). Automated Sleep Apnea Detection in Raw Respiratory Signals using Long Short-Term Memory Neural Networks. *IEEE journal of biomedical and health informatics*.

Van Steenkiste, T., Groenendaal, W., Ruyssinck, J., Dreesen, P., Klerkx, S., Smeets, C., de Francisco, R., Deschrijver, D., and Dhaene, T. (2018b). Systematic Comparison of Respiratory Signals for the Automated Detection of Sleep Apnea. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 449–452. IEEE.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., and Xu, W. (2016). CNN-RNN: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294.

Yin, W., Kann, K., Yu, M., and Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923*.

Young, T., Peppard, P. E., and Gottlieb, D. J. (2002). Epidemiology of obstructive sleep apnea: a population health perspective. *American journal of respiratory and critical care medicine*, 165(9):1217–1239.

Zaffaroni, A., Chazal, P., Heneghan, C., Boyle, P., Ronayne Mppm, P., and Mcnicholas, W. (2009). Sleep Minder: An Innovative Contact-Free Device for the Estimation of the Apnoea-Hypopnoea Index. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2009:7091–4.

Zaffaroni, A., Kent, B., O'Hare, E., Heneghan, C., Boyle, P., O'Connell, G., Pallin, M., de Chazal, P., and McNicholas, W. T. (2013). Assessment of sleep-disordered breathing using a non-contact bio-motion sensor. *Journal of Sleep Research*, 22(2):231–236.

Zhao, M., Yue, S., Katabi, D., Jaakkola, T. S., and Bianchi, M. T. (2017). Learning sleep stages from radio signals: A conditional adversarial architecture. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4100–4109. JMLR. org.

Bjørgan, M. T. and Ulvøen, T.

Classifying Hypopnea and Obstructive Sleep Apnea with the Somnofy Doppler Radar

NTNU
Norwegian University of
Science and Technology

vital things