Sleep detection with photoplethysmography for wearable-based health monitoring

Master of Science Thesis
University of Turku
Department of Computing
Medical Analytics and Health IoT
2021
Susanna Landström

Supervisors:
Ph.D. Iman Azimi
Ph.D. Pasi Liljeberg

UNIVERSITY OF TURKU
Department of Computing

SUSANNA LANDSTRÖM: Sleep detection with photoplethysmography for wearable-based health monitoring

Master of Science Thesis, 52 p.
Medical Analytics and Health IoT
May 2021

Remote health monitoring has gained increasing attention in the recent years. Detecting sleep patterns provides users with insights on their personal health issues, and can help in the diagnosis of various sleep disorders. Conventional methods are focused on the acceleration data, or are not suitable for continuous monitoring, like the polysomnography. Wearable devices enable a way to continuously measure photoplethysmography signal. Photoplethysmography signal contains information on multiple physiological systems, and can be used to detect sleep patterns. Sleep detection using wearable-based photoplethysmography signal offers a convenient and easy way to monitor health.

In this thesis, a photoplethysmography-based sleep detection method for wearable-based health monitoring is described. This technique aims to separate wakefulness and asleep states with adequate accuracy. To examine the importance of good quality data in sleep detection, the quality of the signal is assessed. The proposed method uses statistical and heart rate based features extracted from the photoplethysmography signal. Using the most relevant features, various supervised learning algorithms are trained, compared and evaluated. These algorithms are logistic regression, decision tree, random forest, support vector machine, k-nearest neighbors, and Naive Bayes.

The best performance is obtained by the random forest classifier. The method received an overall accuracy of 81 percent. It was able to detect the sleep periods with 86 percent accuracy and the awake periods with 74 percent accuracy. Motion artifacts occurring during the awake time caused distortion to the signal. Features related to the shape of the signal improved the accuracy of sleep detection, since signal distortion was associated with the awake time. It is concluded that photoplethysmography signal provides a good alternative for wearable-based sleep detection. Future studies with more comprehensive sleep level analysis could be conducted to provide valuable information on the quality of sleep.

Keywords: classification, health monitoring, photoplethysmography, sleep detection, supervised learning, wearables

TURUN YLIOPISTO
Tietotekniikan laitos

SUSANNA LANDSTRÖM: Sleep detection with photoplethysmography for wearable-based health monitoring

Diplomityö, 52 s.
Medical Analytics and Health IoT
Toukokuu 2021

Viime vuosina etänä tapahtuva terveyden seuranta on saanut yhä enemmän huomiota. Unen tunnistaminen antaa käyttäjille tietoa heidän henkilökohtaisista terveysongelmistaan ja voi auttaa erilaisten unihäiriöiden diagnosoinnissa. Tavanomaiset menetelmät käyttävät kiihtyvyyteen perustuvaa dataa, tai eivät ole soveltuvia jatkuvaan seurantaan, kuten polysomnografia. Puettavan teknologian avulla fotopletysmografiasignaalin jatkuva mittaus on mahdollista. Fotopletysmografiasignaali sisältää tietoa useista fysiologisista järjestelmistä ja sitä voidaan käyttää unen tunnistamiseen. Puettavan teknologian avulla mitatun fotopletysmografiasignaalin käyttö unen tunnistuksessa tarjoaa kätevän ja helpon tavan seurata terveyttä.

Tässä diplomityössä kuvataan fotopletysmografiaan perustuva unenhavaitsemismenetelmä, joka soveltuu puettavaa teknologiaa hyödyntävään terveyden seurantaan. Tekniikalla pyritään erottamaan hereillä olo ja uni riittävän tarkasti. Signaalin laatu arvioidaan, jotta voidaan tutkia datan laadun tärkeys unen tunnistuksessa. Kehitetty menetelmä käyttää tilastollisia ja sykkeeseen perustuvia ominaisuuksia, jotka on erotettu fotopletysmografiasignaalista. Tärkeimpiä ominaisuuksia käyttämällä erilaisia valvottuja oppimisalgoritmeja koulutetaan, vertaillaan ja arvioidaan. Käytetyt algoritmit ovat logistinen regressio, päätöspuu, satunnainen metsä, tukivektorikone, k-lähimmät naapurit ja Naive Bayes.

Paras tulos saadaan käyttämällä satunnainen metsä -algoritmia. Menetelmällä saavutetaan 81 prosentin kokonaistarkkuus. Uni pystytään tunnistamaan 86 prosentin tarkkuudella ja hereillä olo 74 prosentin tarkkuudella. Hereillä ollessa liikkeestä johtuvat häiriöt aiheuttavat vääristymää signaaliin. Signaalin muotoon liittyvät ominaisuudet paransivat unentunnistuksen tarkkuutta, koska signaalin vääristyminen yhdistettiin hereilläoloaikaan. Tutkimuksen tuloksista voidaan tehdä johtopäätös, että fotopletysmografiasignaali tarjoaa hyvän vaihtoehdon puettavaa teknologiaa hyödyntävään unen tunnistamiseen. Tulevaisuudessa unen eri vaiheita voitaisiin tutkia kattavammin, jolloin saataisiin arvokasta tietoa unen laadusta.

Asiasanat: luokittelu, terveyden seuranta, fotopletysmografia, unentunnistus, valvottu oppiminen, puettava teknologia

Contents

List of Figures

List of Tables

Acronyms

AUC Area Under ROC Curve

DS Deep Sleep

EEG Electroencephalogram

ECG Electrocardiogram

EMG Electromyogram

EOG Electrooculogram

HRV Heart Rate Variability

IBI Inter-beat-interval

KNN K-nearest Neighbors

LED Light Emitting Diode

LS Light Sleep

MDI Mean Decrease of Impurity

OVR One-vs-rest

PPG Photoplethysmography

PSG Polysomnography

REM Rapid Eye Movement

SVM Support Vector Machine

UWB Ultra-wideband Radar

1 Introduction

In the recent years, there has been a growing interest towards sleep monitoring. Why it is important to get enough good quality sleep [1], how can sleep help with recovery [2], how it impacts the overall well-being of humans [1] and how it can be measured using different devices and setups [3] [4] [5] [6]. Inadequate amount of good quality sleep can lead to various health problems and diseases. Sometimes it might not be enough to go to bed early and sleep for 8 hours per night. People suffer from various sleep disorders that reduce the quality of sleep. [7] Sleep apnea patients suffer from decreased airflow or interruptions in breathing during sleep. [8] Detecting these sleep disorders and disturbances is important. Early diagnosis of sleep disorders can prevent the development of other diseases, improve well-being and reduce health problems. [7] [8]

Polysomnography (PSG) is the gold standard for sleep monitoring. PSG is usually conducted in a sleep laboratory and requires educated specialists and special equipment. PSG uses multiple sensors to simultaneously monitor many biological signals. It records electroencephalogram (EEG), electrocardiogram (ECG), electromyogram (EMG) and electrooculogram (EOG). PSG produces accurate data and results, but it is obtrusive to the patient. PSG is measured by attaching multiple sensors to the body. Each sensor is connected to the recording unit by a wire. These wires restrict the movement that might normally occur during sleep. Another issue is that the monitoring is performed in a sleep laboratory, which is not a normal environment for the patients and can affect the sleep. [4] Because PSG measurements are usually done for one or two nights, it does not provide a

feasible way to continuously monitoring sleep.

Activity-based (actigraphy) sleep monitoring has been widely used for many decades in clinical research. [9] Actigraphs are wearable devices, that are usually worn around the wrist or the ankle. Actigraph derives sleep and wake patterns from acceleration data. Thus, actigraphy provides a non-invasive and continuous way to monitor sleep. Actigraphs are also affordable, which has enabled their extensive usage. [10] However, actigraphy has its limitations and restrictions. Actigraphy may fail to recognise afternoon naps as sleep periods. Also times, when the device has not been used, might be falsely classified as sleep, especially if the device has been taken off right before or after bedtime. [11]

Remote health monitoring using wearable devices, such as smartwatches and smart rings, presents an interesting choice for the traditional methods. Wearable devices are light and convenient to use, and more often provide multiple features for the user. New wearable devices are constantly being developed, and more and more people are buying these devices to monitor their health [12]. Utilizing the data that is already being collected by the devices to accurately detect sleep patterns could bring individual health monitoring to a new level.

In this thesis, we have developed a photoplethysmography-based (PPG-based) sleep detection method for remote sleep monitoring. The focus was on the sleep-wake classification. The raw PPG data was filtered using a standard Butterworth bandpass filter and the data was split into 30 seconds epochs for analysis. Statistical and heart rate based features were extracted for each epoch. The most important features were selected by using random forest classifier. Using the selected features, six machine learning algorithms were trained, compared, and evaluated. These algorithms were logistic regression, decision tree, random forest, k-nearest neighbors, naive Bayes and support vector machine.

The proposed method was implemented using a case study, where data was gathered from 46 participants for one week. PPG data was collected with a wearable device, that

had a PPG sensor. Actigraphy was used as the reference method in this thesis.

The structure of this thesis is as follows: This chapter explains the traditional health monitoring methods and the reasoning behind our proposed method. Chapter 2 presents other work done in this area. Chapter 3 presents general information regarding PPG, classification methods and evaluation metrics. Chapter 4 describes the data preparation steps for our analysis. Chapter 5 presents the feature extraction and selection phases. Chapter 6 describes the training of the classification algorithms. Chapter 7 presents the results of this thesis as well as the conclusions.

1.1 Research questions

This thesis addresses the following research questions:

- **RQ1** *Can wearable-based PPG signal be used in sleep detection?*
- **RQ2** *Does quality assessed PPG signal improve the performance of the sleep detection method?*

For *RQ1*, PPG signal based sleep-wake classifier is built and evaluated to determine the usability of wearable-based PPG signal for sleep detection. For *RQ2*, the quality of the PPG signal is assessed. Using two data sets, the first containing the complete signal and the second containing only good quality signal, the performance of the algorithms is compared and evaluated. Explicit answers to these research questions are presented in Chapter 8.

2 Related work

Sleep monitoring has been widely studied in the recent years. Some of the studies have focused on sleep-wake classification [3] [4], and others have done more comprehensive sleep stage analysis, differentiating multiple sleep stages [6] [13], such as light sleep (LS), deep sleep (DS) and rapid eyes movement sleep (REM) in addition to sleep-wake classification. The goal of many studies is to provide continuous sleep monitoring with minimum intrusion [3] [5]. In the studies, they have used for example face video [3], ultra-wideband radar [5] and pulse oximeter [4] to collect the biological signals.

Completely non-intrusive vision based method using convolutional neural networks has been proposed in [3]. In the study, they extracted remote PPG signal from face videos using face-color intensity. The goal was to classify sleep-wake stages in eye-closed situations. To overcome the issue of low temporal resolution of the estimated heart rate as well as substantial noise existing in the estimated PPG signal, dynamic heart rate filtering was applied.

The dynamic heart rate filter is a second order Butterworth bandpass filter with a goal of removing noise but preserving the information about heart rate. HR data acquired with a wearable sensor is used as a reference. Three comparisons are conducted in the study: wearable heart rate vs. camera heart rate, heart rate vs. PPG signals and PPG signals with or without noise reduction filters. The area under ROC curve (AUC) is used as the base for the comparison and evaluation of the proposed method. By using the dynamic heart rate filtering they get better results compared to the static heart rate filtering, since the

bandwidth is adaptive and utilizes the heart rate information. Video-based methods have limited spatial coverage and might not be suited for extensive usage, as they would require a camera to be installed at home. Cameras raise issues related to privacy and safety.

In [4], they acquired PPG signal using a pulse oximeter placed on a fingertip. Statistical features were extracted from the PPG signal and used to train four different supervised machine learning models for sleep-wake classification. The goal of the study was to separate wakefulness and asleep in overnight sleep. The used machine learning methods were cubic and weighted k-nearest neighbors as well as quadratic and medium Gaussian support vector machines (SVM). In the comparison of the accuracy between the classifiers, medium Gaussian SVM exceeded the others. To improve the classification, the use of morphological features together with the statistical features can be studied.

[5] proposes a solution that uses ultra-wideband (UWB) radar, environmental sensor board and PPG sensor to overcome issues like discomfort of PSG. Their goal is to develop an easy and convenient sleep monitoring service for public users. The service should be affordable, provide relatively good accuracy and be as comfortable as possible for the users. UWB radar is used to measure breathing rate, heart rate and movement. PPG is used to complement the weaknesses of UWB radar such as sensitivity to movement, therefore heart rate and movement are also measured with PPG. UWB radar provides contactless way of measuring, has low power consumption, simple hardware structure and high resolution. In the study, they will only validate that the system can accurately measure heart rate and respiratory rate. This was done by comparing the results from the system to the heart rate and respiratory rate counted by the subjects themselves. The system was able to measure the biological signals accurately. However, further research needs to be done to be able to use the system for sleep monitoring.

Validating an automated sleep analysis based on inter-beat-interval (IBI) series obtained from PPG signal was the goal in [6]. They were interested in the possibility of utilizing the PPG sensors used in many wearable devices and PPG's capability of rec-

ognizing inter-beat-intervals. Inter-beat-intervals reflect the changes of the autonomic nervous system. Their analysis aimed on making a distinction between wakefulness and sleep as well as making a separation between different sleep stages such as LS, DS, and REM. For the validation of this study, they used a previously validated sleep diagnostic software based on ECG and heart rate variability (HRV) with some modifications. The software was adjusted to use only the IBI series acquired from the ECG signal. The IBI series obtained from the PPG signal were analysed using their sleep analysis algorithm and its reliability, accuracy and applicability was evaluated. The results were compared to the gold standard PSG manual scoring and show that the PPG based IBI scoring performs well in making a distinction between different sleep stages, but it is not accurate enough separating sleep and wakefulness. By using other features from the PPG signal together with IBI, the separation between sleep and wakefulness might improve.

[8] and [13] propose an algorithm that uses long short-term memory and HRV. In [8], they propose a method that could be used to detect sleep apnea syndrome in a home environment. Apnea causes decreases in peripheral oxygen levels, which leads to changes in the autonomic nervous system. Autonomic nervous system has been shown to have an affect on the HRV. In the study, they used PSG data collected in laboratory environment. HRV features were extracted and used to train long short-term memory neural network that classified subjects as healthy or potential apnea patients. They received high sensitivity and specificity, but concluded that more extensive research has to be done to validate the model. The model should also be tested with data collected with a device that can be used at home environment. As the next step they presented that the developed model will be integrated in a smartphone application, and the data is collected with a wearable RRI sensor.

[13] performs more comprehensive study and classifies data into 4 classes separating wakefulness and 3 other sleep stages. These sleep stages were REM, light non-REM sleep (N1 and N2) and slow wave sleep (N3). 132 HRV features were extracted from

30 second ECG signal segments. These features were then used to train long short-term memory neural network algorithm. In the study, they received promising results utilizing temporal patterns that occur during sleep. HRV features can be extracted also from other sensors, such as PPG. Therefore, the model could be used with more than one type of data. Further research could be done by using other data acquired unobtrusively to complement the HRV data. The neural network could also be improved through additional learning tasks.

3 Background

3.1 Photoplethysmography

Photoplethysmography (PPG) is an optical method that can be used to discover and measure change in blood volume as it dilates veins and capillaries in subcutaneous tissues [14]. Blood is pumped to the arteries and eventually to the periphery with each cardiac cycle. Ventricular contraction pushes blood to the arteries and makes them dilate. This pressure pulse can be detected, even though the pulse will be attenuated when it reaches the skin.

There are two ways to measure PPG signal, absorption and reflection. The basic idea behind the measurement of PPG signal in both ways is that a light emitting diode (LED) sends light to the skin and a photo diode receives the light. Receiving photo diode detects changes in the absorption or reflection of the light. Depending on the way used to measure the changes in blood volume, the diodes are placed either on opposite sides of the tissue or on the same side. [14]

In clinical settings, PPG is usually obtained with pulse oximeters attached to a patients fingertip. Pulse oximeters measure PPG using light absorption where the diodes are on the opposite sides of the finger. On the device, light emitting diode sends a ray of light from one side of the finger. On the other side, a photo diode receives the light that has travelled through the finger and measures the changes in light absorption. More blood absorbs more light, meaning that on each heartbeat the measured light drops since the increased

blood volume absorbs more light. [14] The pulse oximeter used in clinical settings does not perform well if the blood flow to the periphery is weak [15]. In these cases, the measurements can be done from the head area, like the forehead, ear or nasal septum [14]. PPG can also be measured from multiple places simultaneously which allows exploring disorders in blood circulation. [16]

Smartwatches collect the PPG signal from the wrist using reflection. When PPG is measured in a reflected state, the light is transmitted and received from the same direction. The reflected light changes when the blood volume in the veins change. [14] Blood absorbs more light than the surrounding tissue, so when the blood volume decreases in the veins the reflected light increases and vice versa [17].

Since multiple physiological systems have an impact on the blood flow to the skin, PPG can be used to measure heart rate and HRV, respiratory rate, blood oxygen saturation, blood pressure and hypovolemia or other abnormalities in blood circulation. [14]

3.1.1 PPG signal artifacts

Measuring PPG signal with a wearable device such as a smartwatch can be affected by the ambient light that changes in different environments and day times. These changes in lighting conditions can be seen in the signal especially if the connection between the sensor of the device and skin is not ideal. If the contact between the sensor and skin changes over time, the PPG signal can suffer from motion artifacts. This is very likely when continuously measuring PPG signal with a wearable device, since it is impossible to constrain the movement of the subjects. [14] Noise caused by motion might not be easily removable by conventional filtering methods, since the spectral content overlaps with cardiac signal band [18]. In addition to the lighting and motion artifacts, the quality of the PPG signal can be affected by other factors as well. Skin color, structure of the skin and skin temperature also have an impact on the PPG signal. Dark skin has more pigment that dampens the light more effectively and thus prevents it from penetrating the tissue.

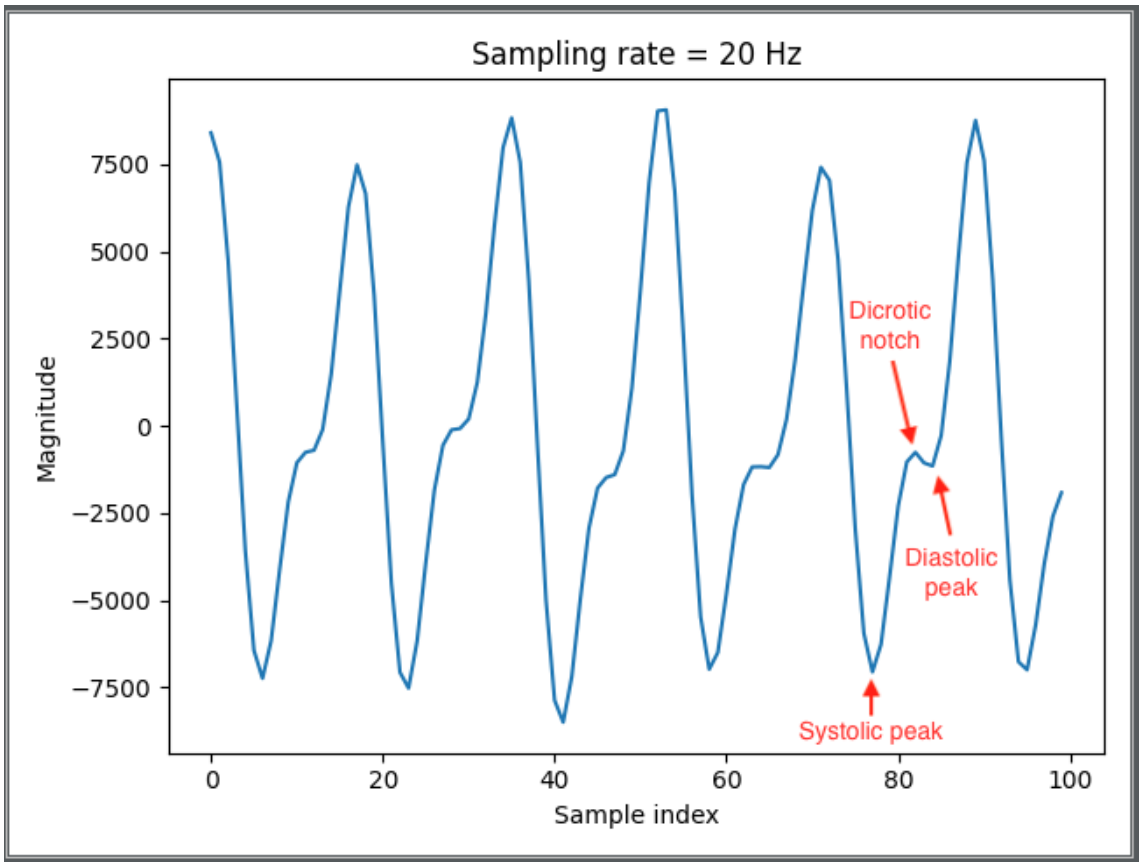

Figure 3.1: PPG signal

These issues are usually dealt with using more powerful light sources or selecting another light frequency. [14]

3.2 Classification

Classification is a supervised learning technique. In supervised learning, an algorithm is taught by giving it pairs of input and a desired output for that input. The algorithm learns that certain patterns in the data are related to a specific output. In classification, the class labels are discrete, for example *true* and *false* or *0* and *1*. The simplest classification problem is a binary classification, where the number of classes is two. [19]

In order to build a proper classification model, the data has to be split into training and testing sets. Typically the data is split so that 70 or 80 percent is used for training the

model. The remaining data is used for testing the performance of the model. [19]

3.2.1 Classification methods

Six machine learning algorithms were trained and evaluated in this thesis. The selected algorithms are some of the most common ones, and they are all suitable for classification task. In the following, the algorithms are briefly described.

Logistic Regression

Logistic regression is a popular statistical model which is mainly used for classification. It predicts the probability of a discrete value, which in our case is sleep or awake. Logistic regression fits an S-shaped logistic function called *Sigmoid function* to the training data using maximum likelihood as demonstrated in Figure 3.2. [19] In the figure, the circles represent observations from class 1 for the training data, and the diamonds on the other hand represent class 0. For each observation we get a probability between 0 and 1 of it belonging to a certain class.

Logistic regression can be used for both binary classification or multi-class problem with ordered or unordered classes. In multi-class problem it is usually used in a way that one class is compared to the rest of the classes. [20] Logistic regression is computationally inexpensive and can be used for large data sets [21].

Decision Tree

Decision tree is a supervised learning method often used for classification. It can be used for both discrete and continuous variables. It works by continuously splitting the data into two or more sub-populations based on some feature or metric. The complete data set is called a root node and first it is split into two or more nodes. If a node is still split into other nodes, the node is called a decision node. If the node is the last node, and there are no arrows pointing away from it, it is called a leaf node. Decision tree helps to determine

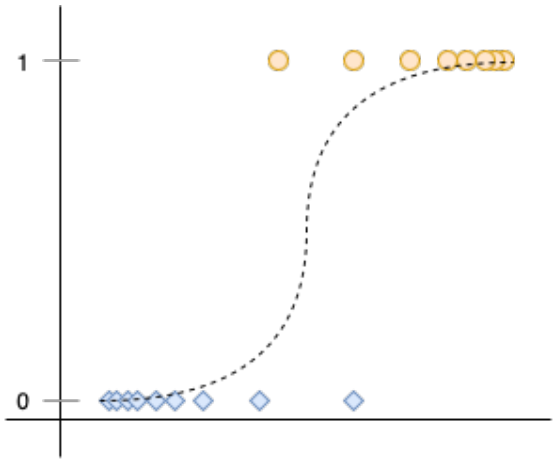

Figure 3.2: Logistic Regression.

the best feature or metric to make the split in each case by considering all possibilities and selecting the one that has the highest information gain. In the ideal case, all the leaf nodes would be pure. A pure leaf node contains observations from a single class. [22]

One example could be that we have data for basketball throws that are either successful or not, meaning that they produce scores or not. As features we have the information whether the player who tossed the ball was a professional or amateur as well as the distance between the hoop and the player. The data is presented in Table 3.2.1. This data is used to build a decision tree that classifies observations which is shown in Figure 3.3. Decision nodes are marked with a red color and the leaf nodes are marked with green color. In this case all the leaf nodes are pure, since they all have observations from only one class. The small circles and diamonds represent the data classes. The circles indicate that a throw has been successful, and the diamonds present the unsuccessful throws.

Random Forest

Random forest is a popular and frequently used supervised learning algorithm, and it can be used for both classification and regression problems. Random forest is combined of multiple decision trees. The basic principle of the random forest is that each decision tree gives a class label to a sample, and the forest chooses the class that is the most common

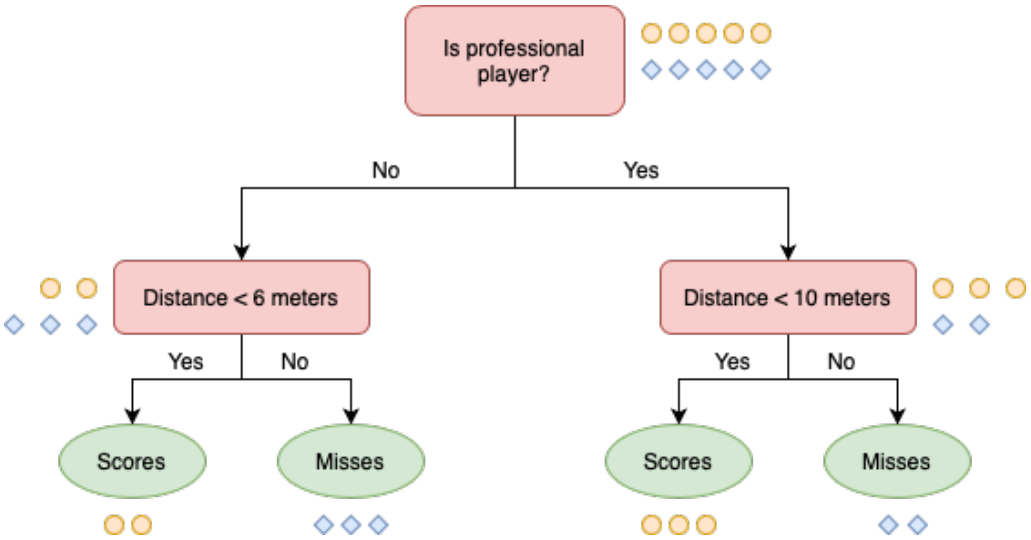

Figure 3.3: An example of a decision tree.

Professional	**Distance**	**Scores**
Yes	5	**Yes**
Yes	7	**Yes**
Yes	10	**No**
No	3	**Yes**
No	6	**No**
Yes	8	**Yes**
No	4	**Yes**
Yes	11	**No**
No	8	**No**
No	7	**No**

Table 3.1: Data of basket ball throws.

between all the trees. [22] The basic idea is demonstrated in Figure 3.4.

Random forest models provide good and reliable results and reduce model's overfitting if the number of decision trees in the forest enough. Random forests are more robust and provide usually better results compared to decision trees. However the computational

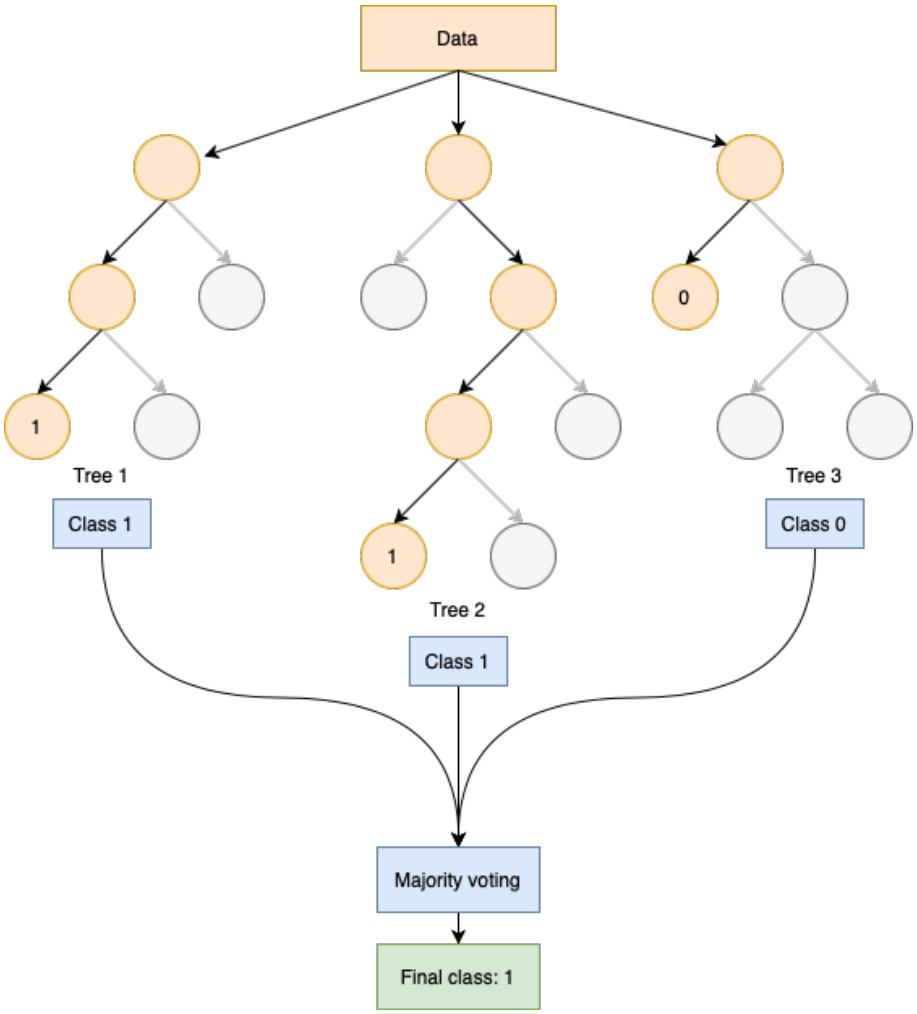

Figure 3.4: Random forest.

time is much higher, which might limit their usage in real-time applications. [23] Random forests can also be used in feature selection, as was done in this thesis. Using random forest in feature selection is described in chapter 5.2.

Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm, which is mostly used for classification tasks but is also suitable for regression problems. The basic idea behind the SVM is that each data point is represented in an *n*-dimensional space, where *n* is the number of features. The goal is to find the most optimal *(n-1)*-dimensional hyperplane that separates the two classes which for us are *asleep* and *awake*. [24] In our case we have 8 features selected and used for the complete data set and 13 features for the quality checked data set. This means that we have 8 and 13 dimensions for these data sets respectively.

In the most simple case where there are only two features and thus 2 dimensions, the hyperplane needed to separate the data points belonging to different classes is a line. There are multiple possibilities for hyperplanes and the most suitable hyperplane is the one that is the furthest away from the data points of both classes. This will give more confidence that new data points are classified correctly. [24] Figure 3.5 indicates multiple possible hyperplanes marked with a dashed line to separate the two classes in a 2-dimensional space. The black dashed line presents the most suitable line since it maximises the distance from both classes.

The data points, which are the closest ones to the hyperplane and therefore affect the position and orientation of the hyperplane, are called support vectors. The SVM is built using the support vectors. Removing the data points that act as support vectors will influence the hyperplane and also the SVM itself. [24]

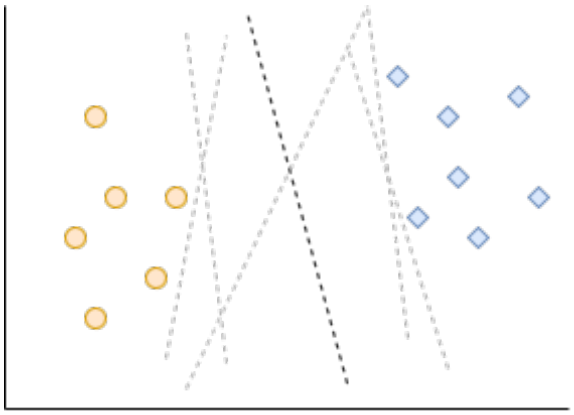

Figure 3.5: Support Vector Machine.

K-Nearest Neighbors

K-nearest neighbors (KNN) method can be used for both classification and regression. [25] KNN works such that all observations are stored and new ones are classified by looking at the nearest data points of the observation. *k* is the number of the closest observations that are used to determine the class for the new observation. The class for the new observation is selected using the majority vote of the k-neighbors, meaning the class that is most presented by its neighbors. For a binary classification problem the number of *k* is usually an odd number to make sure that one class always wins. [26]

Let's say that we have two classes: circles and diamonds. The new observation marked as a triangle should be classified to one of these two classes using k-nearest neighbors method. Let the k be in this case three. Figure 3.6 shows that the three nearest neighbors are all diamonds, so we will classify the new observation as a diamond as well.

Naive Bayes

Naive Bayes is a well-known and frequently-used machine learning algorithm for classification. It is based on a probability theory called Bayes' theorem. Bayes' theorem uses conditional probability to predict the probability of an event or a class. Mathematical equation for Bayes' theorem is presented below. [22]

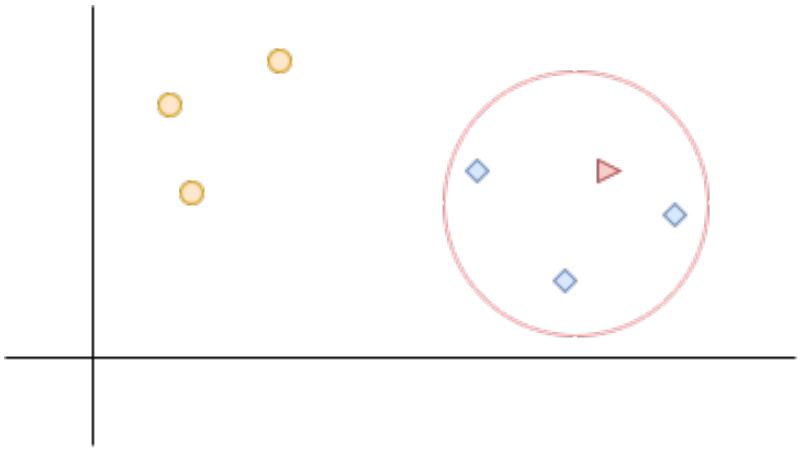

Figure 3.6: K-nearest neighbors.

$$P(c|f) = \frac{P(f|c)P(c)}{P(f)} \tag{3.1}$$

P(c | f) is the probability of class c given that f is the features.

P(c) is the probability of class c independently.

P(f | c) is the probability of features given that class is c.

P(f) is the probability of features.

In Naive Bayes classification, we assume that there is no dependency between the features. We calculate how many times each feature value is presented in each class. For discrete values, it works quite well but continuous variables are assumed to follow normal (Gaussian) distribution which can cause the model to perform poorly if the data is not normally distributed. Another issue is the assumption made about features being independent, since this is rarely the case in the real life. Naive Bayes algorithm performs well with multi-class problems. Training the algorithm is fast, and can be done with a small training set. [27]

3.2.2 Evaluation Metrics

Evaluating the trained classification algorithm is a relevant part of data analysis. Accuracy is the most frequently used and well-known metric in evaluation. However, it might not always tell the whole truth. If the data set is highly imbalanced, the algorithm can receive very high accuracy by classifying each observation belonging to the majority class. Therefore, it is best to evaluate the performance of the algorithm by using multiple metrics. [28] The metrics used in this study are presented below.

Confusion matrix

Confusion matrix is a tool that visualizes the performance of a machine learning algorithm and provides useful statistics that can be used to calculate other metrics as well. Confusion matrix is used in supervised learning and classification. In binary classification, the observations are classified to one of two classes. In our case, the classes are *awake* and *asleep*. The confusion matrix has therefore two rows for the actual values and two columns for the predicted values. [29] These constitute a matrix that is presented in Table 3.2.2.

	Predicted Awake	**Predicted Asleep**
Actual Awake	TN	FP
Actual Asleep	FN	TP

Table 3.2: Confusion matrix

TP is the number of true positives

Predicted as sleeping and they sleep

TN is the number of true negatives

Predicted as awake and they are awake

FP is the number of false positives

Predicted as sleeping and they are awake

FN is the number of false negatives

Predicted as awake and they sleep

Accuracy

Accuracy is the most well known metric when talking about the performance of an algorithm. It is the ratio of the number of times the model predicts correctly to the number of all the predictions. [28]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

Recall

Recall or sensitivity is the true positive rate, which indicates how well the model will identify that the person is asleep. The number of correctly predicted positive results is divided by the number of results that should have been classified as positive, meaning the true positives and false negatives. Recall is calculated with the following formula. [28]

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

Specificity

Specificity is the true negative rate, showing how accurately the negative class has been predicted. In our case, this means how accurately the algorithm will identify that the person is awake. The number of correctly predicted negative results is divided by the number of results that should have been classified as negative, meaning the true negatives and false positives. Specificity can be calculated using the following formula. [30]

$$Specificity = \frac{TN}{TN + FP} \tag{3.4}$$

Precision

Precision determines how reliable the algorithm is. It shows the number of items correctly classified as the positive class. In this study, it indicates the number of cases where the algorithm has correctly labeled the sample as asleep. [28]

$$Precision = \frac{TP}{TP + FP} \tag{3.5}$$

F1-score

F-score is used in statistical analysis of binary classification to calculate the algorithms accuracy using recall and precision values. F1-score is the harmonic mean of precision and recall. The highest possible score is 1.0 and the lowest is 0. The highest score is achieved with perfect recall and precision values, and the lowest if either of the recall or precision is zero. [28]

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{3.6}$$

Mean absolute error

Mean absolute error indicates how much the predicted classes produced by the machine learning algorithm differ from the correct classes as an average. Absolute error can be calculated with the formula below. [28]

$$MeanAbsoluteError = \frac{1}{N} \sum_{j=1}^{N} |y_j - \hat{y}_j| \tag{3.7}$$

4 Data preparation

Data preparation is the process of collecting data and examining the data to find inconsistencies, missing data and anomalies. Possible errors in the data are corrected, and the parts of the data that are relevant for a particular analysis are selected [31].

4.1 Data collection

Data collection is a method of gathering and measuring information. There are several ways to collect data. It can be done by the means of surveys or questionnaires, interviews, observing or physically measuring it with some type of devices. Data collection occurs after defining the research problem and deciding what data will be needed and how to best collect it. [32]

The data used in this thesis was part of the data collected in a case study [33], which was conducted by the Digital Health Technology group, Department of Computing, University of Turku [1]. The study had 46 participants, and measurements were taken continuously during the monitoring period of one week. For this thesis, we used data from two devices, a wearable device and and an actigraphy device. Both devices were worn in the wrist of the non-dominant hand. Participants reported their sleeping times, such as in-bed-time, wake-up-time, and possible nap time as well as times when they had taken the devices off the wrist.

The wearable device used in the study was the Samsung Gear Sport watch [34], which

[1] https://healthtech.utu.fi/

has an embedded PPG sensor. In the case study, the watch was programmed to collect both PPG and acceleration data. They had also developed an application that uses Wi-Fi to send the collected data to a data server.

The actigraphy device used in the case study was ActiGraph's wGT3X-BT [35], which measures acceleration data in three dimensions. In this study, actigraphy was selected as the reference method for sleep detection.

4.1.1 PPG in this study

In this study, the raw PPG signal was measured with the Samsung Gear Sport watch. As stated before, PPG measures the changes in blood volume and therefore it can be used to observe biological signals, like heart rate and HRV. Figure 3.1 shows a view of a PPG signal used in this thesis. From the signal we can see that when blood volume increases in the veins, the measured signal is weaker and when the blood volume decreases the signal is stronger. When a person is sleeping, their heart rate decreases. HRV on the other hand increases during sleep. [36] These physiological values in addition to many others can be used to distinct sleep and wakefulness from each other.

To reach the goal of differentiating between sleep and awake stages, features were extracted from the PPG signal. Part of these features were simple statistical features, but others utilized the heart rate and HRV.

HRV means how the heart rate changes over time. The basis for many HRV parameters is the intervals between normal heart beats, also called NN intervals. The most common HRV parameter is SDNN, which is the standard deviation of all NN intervals. Normally SDNN is calculated for the entire recording, but can also be calculated for shorter time periods. [37] In this thesis, the SDNN parameter was calculated for each 30 second epoch. rMSSD, pNN20 and pNN50 are parameters that measure how HRV changes from beat to beat. The abbreviation rMSSD comes from root mean square of successive differences, and it indicates how the inter-beat-interval between beats changes

on average for one epoch. pNN20 and pNN50 describe the percentage of all beats where the difference from one beat to the next beat is more than 20 or 50 seconds respectively. [37]

4.2 Data discovery and profiling

In data discovery and profiling, we are examining the data to understand its structure, the sampling rate, the errors, and the missing values. [31] In this step we choose the part of the data that we will be using in this thesis.

The watch data consists of PPG, acceleration in three dimensions and timestamps. The actigraphy data had only acceleration in three dimensions and timestamps. The sampling rates were 20 Hz for the watch and 80 Hz for the actigraphy. The actigraphy data had several errors, when it was compared to the self-report data from the participants. When comparing the acceleration data from both devices, we noticed that they were not in sync. Figure 4.1 demonstrates the out of sync acceleration data, which needed fixing.

4.3 Data cleansing

Data sets might have missing data values or erroneous data. In data cleansing, the errors or abnormalities are corrected or removed from the data set [31]. An example of correcting erroneous data could be fixing typing errors in surveys used in data analysis. It is also possible to add missing values by combining two data sets to compensate the missing values of the original data set.

The errors in the actigraphy data were corrected using the self-reported reports. The timestamps of the watch data and actigraphy data were synced using a cross-correlation technique. The acceleration data from both of these devices was used to find the shift between the timestamps. A shift value was calculated separately for each subject and used to calculate new timestamps for the watch data.

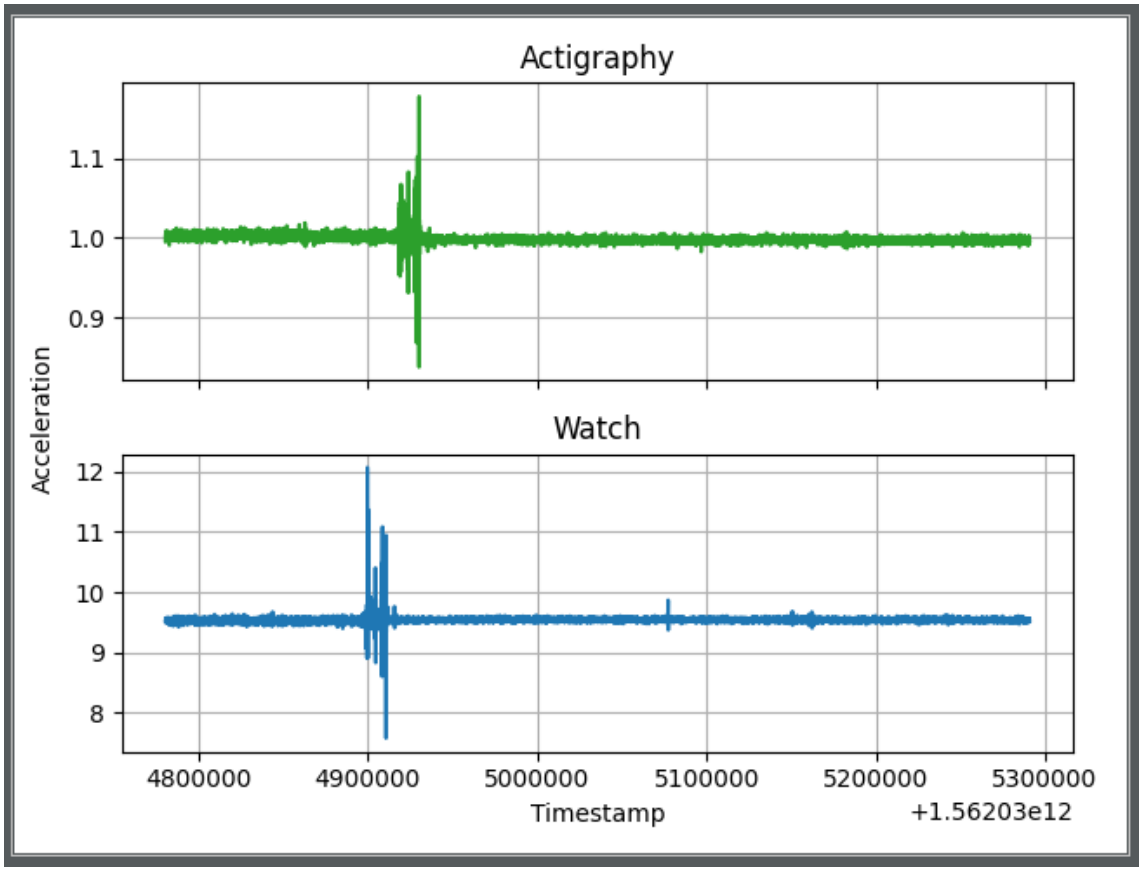

Figure 4.1: Out of sync acceleration data from the actigraphy (green) and the watch (blue).

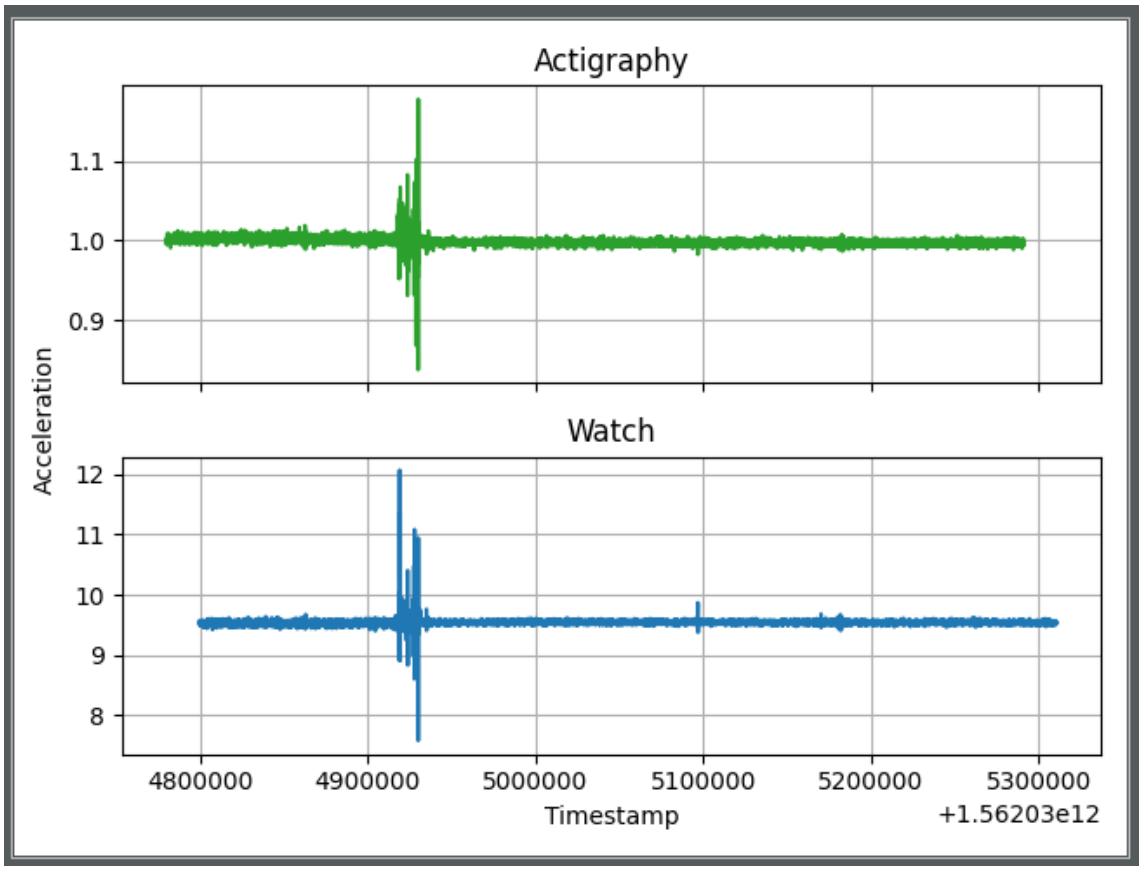

Figure 4.2: Synced acceleration data from the actigraphy (green) and the watch (blue).

Figure 4.1 shows both the actigraphy and the watch acceleration data before shifting the timestamps. In Figure 4.2, the timestamps of the watch data have been shifted with the calculated value. We can see that the acceleration data for these two devices is now aligned.

Some subjects were excluded from the analysis since their data were incomplete. A few subjects had missing PPG recordings due to the fact that the device was not used. A few other subjects had missing data related to the time they had evidently used the devices. This data is important in this study since the times were used as one basis for labeling. As the result, we ended up using data collected from 42 subjects.

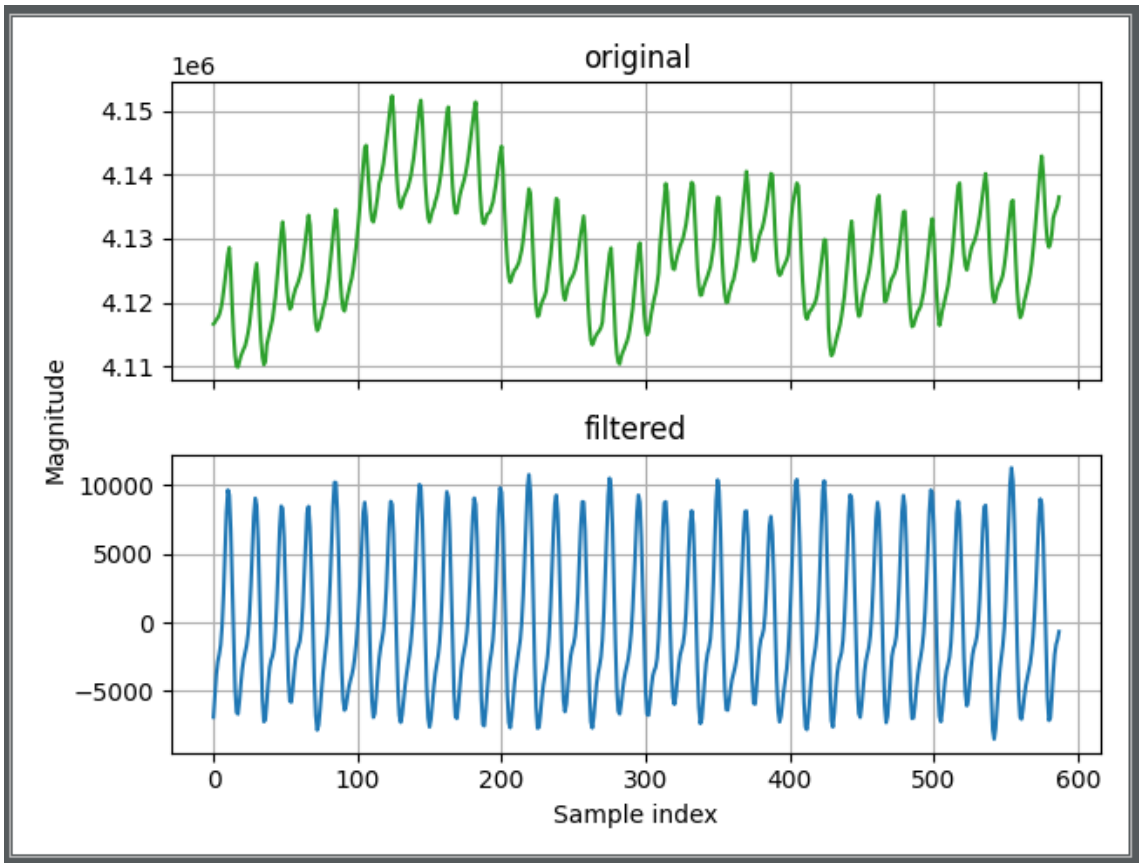

Figure 4.3: PPG signal before (green) and after (blue) filtering.

4.4 Filtering

The PPG data was filtered in order to remove possible noise. We used a standard Butterworth bandpass filter with cutoff frequencies of 0.6 Hz and 3.0 Hz. These cutoff frequencies were selected because we only wanted to include frequencies that could contain information about the heart rate. 0.6 Hz and 3.0 Hz correspond to 36 bpm and 180 bpm respectively.

Figure 4.3 shows PPG signal prior to filtering and after the signal has been run through a Butterworth filter.

4.5 Labeling

Data labeling means that images, videos, data points etc. are identified and marked with labels. This is necessary for implementing supervised machine learning methods for a data set. [38]

Two steps should be considered, when determining a label (i.e., asleep or awake) to a data point. The first step was the inclusion of the trial period: i.e., the time when the subject used the devices. Any data point outside this trial period would be labeled as *none* and would not be used in the later stages of the analysis. The second step was the inclusion of the sleep periods collected from the automatic sleep detection of the Actigraph. The sleep periods have values for in bed time and out bed time for each night, so the data points with timestamps between some of these values were labeled as *asleep*. Data points outside the sleep periods were labeled as *awake*.

4.6 Splitting into epochs

For feature extraction and further steps in the analysis, the data has to be split into epochs of suitable length. Single data points cannot be used since we are analysing a signal. We used epochs of thirty seconds in this thesis. After the data had been split into epochs, the epochs were evaluated. Only epochs that contained data points with the same label were included in the analysis. If epochs contained both labels for sleep and awake, they were discarded.

4.7 Quality assessment

The signal quality was assessed to investigate whether the results could be improved by selecting PPG signal with only good quality to be used in the analysis. Quality assessment method from [39] was used to extract features from the data. The extracted features were

used with a support vector machine algorithm to classify the epochs as either good quality or bad quality. Only the good quality epochs were included in the quality checked data set.

As a result, the number of epochs was reduced by 54 percents from 197 500 epochs to 90 966 epochs. Another outcome was that 83 percent of the epochs passing the quality inspection were labeled as asleep. This was not surprising, and it supports the fact that it is easier to extract good quality data at rest. While the subjects are awake, they might engage in various activities, with the extremities being exercising and laying on the couch. As stated in chapter 3.1.1, noise caused by motion can not be filtered out using standard filtering techniques. This explains why the epochs measured while the subjects are awake are more often discarded.

5 Feature Extraction and Selection

5.1 Feature extraction

In feature extraction, combinations of the variables in the original set of raw data are calculated. These combinations are the features that describe the original data set. Feature extraction reduces the dimensionality of the original set of raw data by still preserving the characteristics of the data. By reducing the dimensionality of the data, feature extraction makes the machine learning process faster and computationally less expensive. [40]

In the feature extraction phase conducted in this thesis, each thirty second epoch is handled separately and features are extracted for that specific epoch. Each epoch also has a label that is needed when training the classification algorithms. In our case, the epochs were 30 seconds long and our sampling frequency was 20 Hz. This means that each epoch includes 600 samples of PPG. The total number of features extracted in this thesis was 32. Therefore, when we extract features from each epoch, we reduce the dimensionality from 600 to 32.

In this thesis, several types of features were extracted from the PPG data. First statistical features in time domain were extracted from the PPG data. Second, PPG signal's Power Spectral Density (PSD) was calculated using Welch's method, and statistical characteristics of the PSD were extracted. For the PSD, some additional features were also calculated, such as spectral entropy, mean frequency and median frequency.

Third, features related to biological properties are extracted. In this regard, we used

a Python Heart Rate Analysis Toolkit called HeartPy. HeartPy has been developed to analyse noisy PPG signal collected with PPG sensors, smartwatches and smart rings in experimental studies [41] [42].

Two types of measures were extracted from a heart signal: i.e., measures related to heart rate and measures related to HRV. The most well known heart rate measure is beats per minute (bpm), which is also called a pulse. Another measure for heart rate is inter-beat-interval, which is the time interval between two heart beats. HRV measures the changes of the inter-beat-intervals over time. [41] [42] The list of features extracted is presented in the following.

1. Statistical features extracted in time domain as well as frequency domain.
    - Mean absolute deviation
    - Interquartile range
    - Twenty-five trimmed mean
    - Max value
    - Min value
    - Skewness
    - Kurtosis (Fisher)
2. Additional features extracted in frequency domain.
    - Geometric mean
    - Harmonic mean
    - Spectral entropy
    - Mean frequency
    - Median frequency
3. Heart rate and heart rate variability features extracted.

- Beats per minute (BPM)
- Interbeat interval (IBI)
- Standard deviation of NN intervals (SDNN)
- Standard deviation of successive differences (SDSD)
- Root mean square of successive differences (rMSSD)
- The proportion of NN20 divided by total number of NNs. (pNN20)
- The proportion of NN50 divided by total number of NNs. (pNN50)
- Heart rate median absolute deviation (HR MAD)
- Poincaré plot standard deviation perpendicular the line of identity (SD1)
- Poincaré plot standard deviation along the line of identity (SD2)
- Area of the ellipse which represents total HRV (S)
- Ratio of SD1-to-SD2 (SD1/SD2)
- Breathing rate

Features were extracted for two sets of data. First, feature extraction was performed on the original data set that contained all epochs. Second, features were extracted from the data that had been quality checked. Therefore, we have two set of features that are used separately in the next steps. The data set containing all epochs is later referred as the *complete data set*, and the data set composed of the quality checked epochs is called the *quality checked data set*.

5.2 Feature selection

In feature selection, the most relevant features that best describe the data are selected. The goal is to identify redundant and irrelevant features, which can be removed without causing loss of information. Redundant features are those that in itself are relevant to

describe the data, but that correlate strongly with another relevant feature. Thus, the redundant feature can be removed since the correlating feature is used for the modeling. [43]

There are many advantages of selecting only the most essential features to be used in the training of the machine learning algorithms. This step reduces the dimensionality of the data even further. With fewer features, the training of the algorithms becomes faster, meaning that the time needed to train the algorithms is shorter, and less computational capacity is required. It is also easier to interpret the results of the algorithm, when fewer features are used. A machine training algorithm trained with fewer features might perform better when tested with an unknown set of data. If the algorithm is trained using too many features, it might become overfitted which means that it presents the data set used in training too well and might include features of noise in the data set. Overfitting causes an algorithm to perform poorly when used on a new set of data. [43]

Feature selection can be done manually or automatically. [43] In manual feature selection, humans evaluate the correlations between features, and select the ones that best describe the data. This is possible if the number of features is not too large. With large data sets, where many features are extracted, it is much more efficient to use automatic feature selection. In automatic feature selection, the importance of each feature is calculated using an algorithm, and only the ones that score higher than predefined threshold are selected. The threshold can be determined arbitrary, or it can be calculated for a specific case by for example using the importance scores for all features and taking the mean value as the threshold.

Random Forest was used as the method to select features, which were used to train the machine learning algorithms. Random Forest gives the importance of each feature. The importance indicates how much that feature affects the decision. The importance is based on the mean decrease of impurity (MDI). For classification, Gini index is used as the measure to determine the reduction of the impurity. [44] Random forests are very

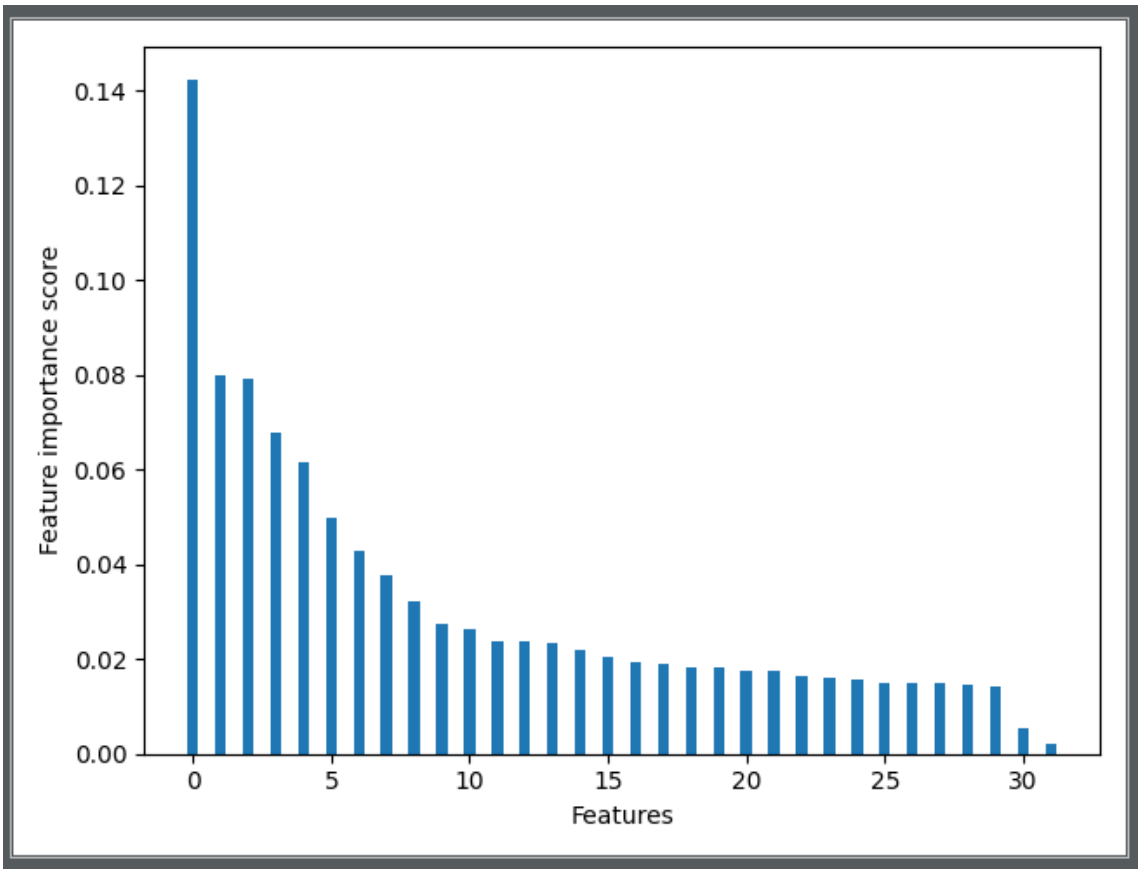

Figure 5.1: Histogram of features from complete data set sorted by their importance score.

popular method to select features because the results are easy to interpret, and they also provide high accuracy and low overfitting. [45]

In this study, we utilised Sklearn's Random Forest Classifier with the number of trees being a hundred. Before using the machine learning algorithm, the data was split into training and testing sets with 70-30 split respectively. Only the training set was used for feature selection. Features with importance score higher than the mean importance of all features were selected. [45]

Figure 5.1 shows the features extracted from the complete data set sorted by their importance score. In this case the threshold score was 0.0312, so the features in Table 5.1 were selected.

Figure 5.2 shows the features extracted from the quality checked data set sorted by

Feature	**Score**
Kurtosis	0.1422
PSD Spectral Entropy	0.08
Inter-beat-interval	0.0793
PSD Skewness	0.0679
Beats per minute	0.0617
25 Trimmed Mean	0.0499
PSD Kurtosis	0.0428
Skewness	0.0378
Min Value	0.0321

Table 5.1: Selected features for the complete data set.

their importance score. Threshold in this case was 0.0313, and so the features in Table 5.2 were selected.

When comparing the histograms for both data sets, we can notice that there is more variance between the features in the complete data set. With more variance, it is easier to select the most important features to be used in the classification algorithms. Therefore, we could assume that the final results would be better for the complete data set.

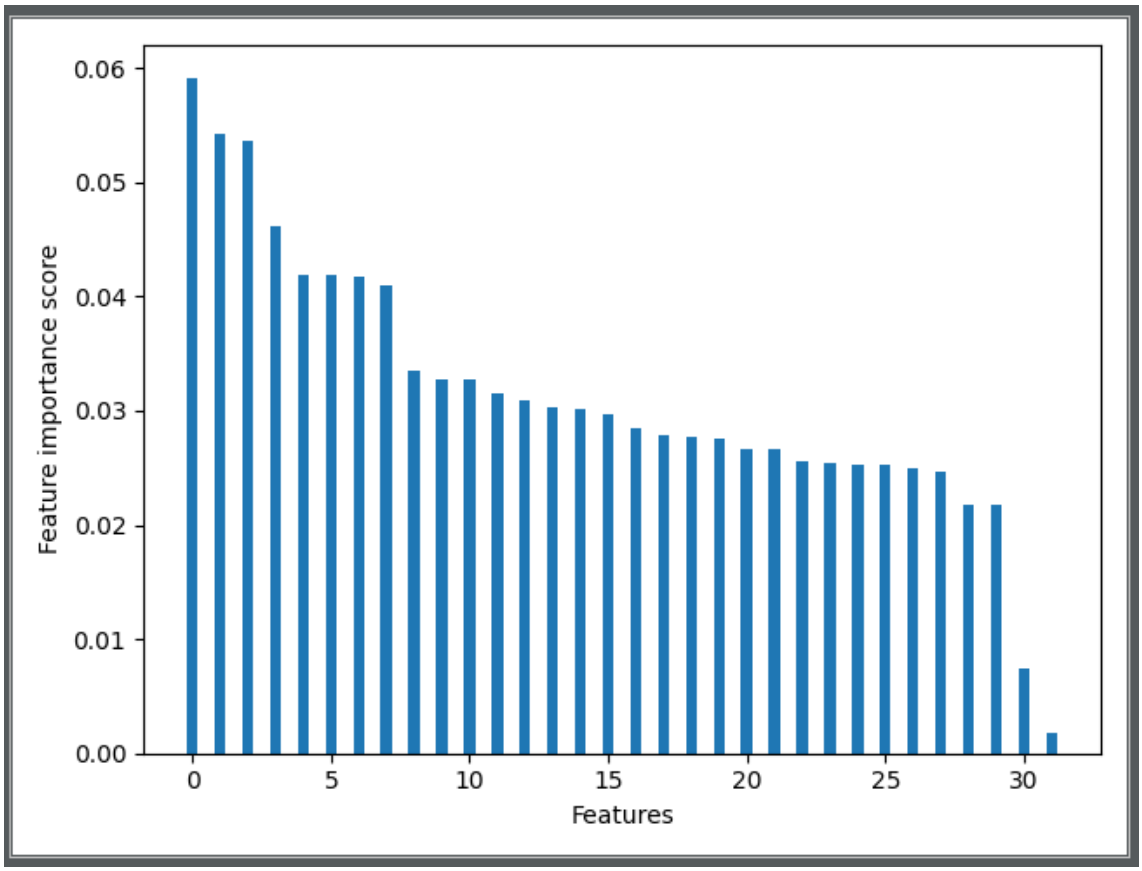

Figure 5.2: Histogram of features from quality checked epochs sorted by their importance score.

Feature	**Score**
Inter-beat-interval	0.0591
Beats per minute	0.0542
PSD Mean Frequency	0.0536
Skewness	0.0461
Kurtosis	0.0419
Breathing rate	0.0419
25 Trimmed Mean	0.0417
PSD Median Frequency	0.041
Interquartile range	0.0334
PSD Mean Absolute Deviation	0.0328
Mean Absolute Deviation	0.0327
Min Value	0.0316

Table 5.2: Selected features for quality checked data set.

6 Training classification algorithms

For the machine learning part in this thesis, features have already been extracted from each 30 second epoch. Each line in the data set presents one epoch and its features. The first step was to see if there were any missing values in our data. Lines containing missing values were removed from the data set. The next step was to get a grasp of how balanced our data is, meaning the ratio between data samples labeled as *asleep* and *awake*. If there is a significant difference between the number of each label presented in the data, the data set has to be balanced in order for it to be usable in the classification methods.

The data set can be balanced using up-sampling or down-sampling. In an imbalanced data set, one of the classes is more common than the other class. The class that has more observations in the data set is called the majority class. The more infrequent class is called the minority class. Up-sampling means that some of the minority class observations are randomly duplicated. Down-sampling on the other hand is done so that the majority class is randomly reduced by removing observations. The goal in both cases is to make the classes evenly presented so that both classes have the same amount of observations. [46]

For training different machine learning models, the data were split into training and testing sets. The separation was done randomly and so that 70 percent of the data was used for training and the remaining 30 percent was used for testing. The test set can be also called hold-out set which means that the data in the hold-out set is something that the model has not seen before. Therefore, by testing the model with the hold-out set, we can see how the model performs with unseen data. [47] The training and testing sets do

not share users. Raw feature values were standardized before the data was used with the machine learning models. Standardization makes the features normally distributed with zero mean and unit variance, so each feature has an equal effect to the decision. Scikit learn's StandardScaler from their preprocessing module was used to standardize the data set. [48]

All six classification algorithms were trained using the training set and their performance was tested with the testing set. For each algorithm, we created a pipeline, which conducted standardization of the data before fitting the data to the model. Each of these steps was done for both the complete data set as well as the quality checked data set.

6.1 Complete data set

The complete data set includes 197500 30-second epochs. When calculating the missing values for the data, there were several missing values in the heart rate and HRV features. These lines were removed from the analysis. After this step the number of lines was reduced to 162 769. To determine the balance of the data set, we calculated how each label was represented in the data. For the complete data set, 54 percent of the labels were *asleep*. The balance was good enough, so no further steps were needed in this case.

The data was split into training set and testing set. After the split was done, we were able to see how the labels were presented within each set. The split was very even, in both sets 53.7 percent of the labels were *asleep*, which was also consistent with the balance of the complete data set. Only features selected using the random forest classifier were used to create the machine learning models. These features are listed in the previous chapter in Table 5.1.

6.2 Quality checked data set

The quality checked data set includes 90 966 30-second epochs. When searching for missing values from this data set, we found that there were multiple missing values among the heart rate and HRV features. This was expected, since the quality checked data set is a portion of the complete data set. Each line containing missing values was removed from the data set and as the result we had a data set with 85 424 lines.

When examining the balance of the data set, we noticed that 83 percent of the data was labeled as *asleep*. Both up-sampling and down-sampling were tested to see how each of them affected the results. In this case, up-sampling produced better results. Up-sampling was only conducted on the training data.

After the quality checked data had been split into training and testing sets, we examined how each label was presented in each of these sets. The split had been even, and both labels were equally presented in both sets. Like with the complete data set, only the selected features were used for training the machine learning models. The features used for the quality checked data set are listed in Table 5.2.

6.3 Used models

The classification algorithms presented in section 3.2 were trained using the settings and parameters described below. The algorithms were implemented using scikit-learn [49].

6.3.1 Logistic regression

In our case, the number of samples was greater than the number of features. This is why dual formulation was set false. *Lbfgs* solver was used in the optimization problem with *L2* penalty. Multi class value was set to auto, which chooses *ovr* (one-vs-rest), since our data is binary.

6.3.2 Decision tree

In the decision tree, at each node, the best split was chosen based on gini impurity. As the maximum depth of the tree we used the value *None*, in which case the nodes are split until each leaf node is pure. The number of leaf nodes was not limited, so the value for max leaf nodes was *None*.

6.3.3 Random forest

In our forest, the number of trees was 100. As with the decision tree, in random forest gini impurity was used as the criteria to measure the quality of each split. Maximum depth was not set, so the splits were done as long as the leaves reached purity. The number of leaf nodes was also not limited.

6.3.4 K-nearest neighbors

The number of neighbors in our k-nearest neighbors classifier is set to 5. The weights parameter value is *uniform*, meaning that each neighbor point affects the decision with equal weight. As the metric to determine the closest neighbors, we use parameter value *minkowski*. Together with power parameter value 2, the distance metric is equal to standard Euclidean metric.

6.3.5 Naive Bayes

Gaussian Naive Bayes algorithm was used with no prior probabilities. Priors were adjusted based on our data, since they were not preset.

6.3.6 Support vector machine

Support vector machine (SVM) was used with linear kernel. *Dual* parameter can be chosen for the algorithm to solve either dual or primal optimization problem. In our case, the

number of samples is larger than the number of features. Therefore, the dual parameter was set to false.

7 Results

Interpreting and evaluating the trained classification algorithms is the final step of data analysis [32]. The results of our analysis are presented and evaluated using the metrics described in section 3.2.2.

7.1 Results for the complete data set

Confusion matrices for the classification algorithms trained with the complete set of data are presented in Tables 7.1 - 7.6. When looking at the confusion matrices, we can notice that the numbers for logistic regression, random forest, kNN and SVM are quite similar. For the decision tree, the number of false positives and false negatives are higher, and the number of true positives and true negatives are lower. The false positives of the Naive Bayes are significantly higher compared to the other models, and the number of false negatives is the lowest. Therefore, we can assume that decision tree and Naive Bayes will not perform as well as the other algorithms.

7.1.1 Model comparison for complete data set

Tables 7.7 and 7.8 show the evaluation metrics for each of the algorithms. The tables are sorted in descending order by precision. As we can see from Table 7.1, Random forest has the highest accuracy of all the classification algorithms. However, as was mentioned in section 3.2.2, we should consider all the different evaluation metrics, when determining

	Predicted Awake	**Predicted Asleep**
Actually Awake	16487	5792
Actually Asleep	3805	24617

Table 7.1: Random Forest's confusion matrix for complete data

	Predicted Awake	**Predicted Asleep**
Actually Awake	15936	6343
Actually Asleep	3953	24469

Table 7.2: Support Vector Machine's confusion matrix for complete data

	Predicted Awake	**Predicted Asleep**
Actually Awake	15934	6345
Actually Asleep	4024	24398

Table 7.3: Logistic Regression's confusion matrix for complete data

	Predicted Awake	**Predicted Asleep**
Actually Awake	16057	6222
Actually Asleep	4520	23902

Table 7.4: K-Nearest Neighbors' confusion matrix for complete data

	Predicted Awake	**Predicted Asleep**
Actually Awake	15534	6745
Actually Asleep	6946	21476

Table 7.5: Decision Tree's confusion matrix for complete data

	Predicted Awake	**Predicted Asleep**
Actually Awake	13680	8599
Actually Asleep	2732	25690

Table 7.6: Naive Bayes' confusion matrix for complete data

Model	Accuracy	Precision	Mean Absolute Error
Random Forest	0.810714	0.809530	0.189286
SVM	0.796927	0.794139	0.203073
Logistic Regression	0.795487	0.793612	0.204513
kNN	0.788130	0.793454	0.211870
Decision Tree	0.729966	0.760994	0.270034
Bayes	0.776513	0.749220	0.223487

Table 7.7: Model comparison for complete data set

Model	Recall	Specificity	F1
Random Forest	0.866125	0.740024	0.836872
SVM	0.860918	0.715292	0.826181
Logistic Regression	0.858420	0.715203	0.824744
kNN	0.840968	0.720724	0.816520
Decision Tree	0.755612	0.697249	0.758293
Bayes	0.903877	0.614031	0.819314

Table 7.8: Model comparison for complete data set

which algorithms has the best performance.

In the terms of precision, accuracy, specificity, F1-score and mean absolute error, we can see that the Random forest has the best performance. Recall was the only metric, in which Random forest did not exceed the all other algorithms. The highest recall value was achieved by Naive Bayes. If we look more closely the confusion matrices, we can notice that the Naive Bayes algorithms predicts the class *asleep* more often than the other algorithms. This explains the high recall value. From the Naive Bayes confusion matrix we can also see that the number of false positives is also the highest. This explains the lowest specificity of the algorithm.

7.2 Results for quality checked data set

Confusion matrices for the classification algorithms trained with the quality checked data are presented in Tables 7.9 through 7.14. When analysing the confusion matrices, we can notice that the number of true negatives is extremely low for the naive Bayes and random forest. This will indicate that the specificity is low and the algorithms are not able to detect the negative class.

7.2.1 Model comparison for quality checked data set

Evaluation metrics for the quality checked data set are presented in Tables 7.15 and 7.16. Tables are sorted in descending order by precision. The best overall performance is achieved by SVM and logistic regression. Random forest, naive Bayes and decision tree have higher accuracy than SVM and logistic regression. However, these algorithms have extremely low specificity, which shows that they are not able to detect wakefulness state. All of the algorithms have relatively low specificity, but SVM and logistic regression are still able to detect wakefulness with 54 percent accuracy.

	Predicted Awake	**Predicted Asleep**
Actually Awake	2343	2001
Actually Asleep	5927	17532

Table 7.9: Support Vector Machine's confusion matrix for quality checked data

	Predicted Awake	**Predicted Asleep**
Actually Awake	2354	1990
Actually Asleep	6035	17424

Table 7.10: Logistic Regression's confusion matrix for quality checked data

	Predicted Awake	**Predicted Asleep**
Actually Awake	2156	2188
Actually Asleep	7729	15730

Table 7.11: K-Nearest Neighbors' confusion matrix for quality checked data

	Predicted Awake	**Predicted Asleep**
Actually Awake	975	3369
Actually Asleep	2033	21426

Table 7.12: Random Forest's confusion matrix for quality checked data

	Predicted Awake	**Predicted Asleep**
Actually Awake	1193	3151
Actually Asleep	3808	19651

Table 7.13: Decision Tree's confusion matrix for quality checked data

	Predicted Awake	**Predicted Asleep**
Actually Awake	401	3943
Actually Asleep	1759	21700

Table 7.14: Naive Bayes' confusion matrix for quality checked data

Model	Accuracy	Precision	Mean Absolute Error
SVM	0.714851	0.897558	0.285149
Logistic Regression	0.711362	0.897497	0.288638
kNN	0.643312	0.877888	0.356688
Random Forest	0.805704	0.864126	0.194296
Decision Tree	0.749703	0.861810	0.250297
Bayes	0.794914	0.846235	0.205086

Table 7.15: Model comparison for the quality checked data set

Model	Recall	Specificity	F1
SVM	0.747346	0.539365	0.815594
Logistic Regression	0.742743	0.541897	0.812819
kNN	0.670532	0.496316	0.760326
Random Forest	0.913338	0.224448	0.888051
Decision Tree	0.837674	0.274632	0.849571
Bayes	0.925018	0.092311	0.883874

Table 7.16: Model comparison for the quality checked data set

7.3 Comparing the data sets

When evaluating the results for both data sets, we can conclude that the algorithms trained with the complete data set performed better. Based on the feature selection done in section 5.2, we predicted that the complete data set would have higher performance. One possible explanation is that the good quality signals for both classes were more similar to each other. Separating the classes is more difficult due to the similarity between features of both classes. As was seen in Figure 5.2, there was not much variance in the feature scores for the quality checked data set.

The complete data set contains distorted data due to artifacts, such as hand movement. In this case, the noisy data works in our advantage, since it serves as a feature making a more clear distinction between sleep and wakefulness states. By conducting the quality assessment and removing the bad quality data, we end up with lower specificity compared to the complete data set. The performance of the method is improved by extracting features related to the shape of the signal from both a good quality signal and a distorted signal.

8 Conclusion

In this thesis, we examined whether wearable-based PPG signal can be used in sleep detection, and whether the results can be improved by only using quality assessed signals. We proposed a sleep detection method using PPG data acquired with wearable devices. In the analysis, the signal was filtered and segmented into 30-second epochs. Quality assessment was conducted to the original data set, and as the result we analysed two data sets: i.e., the complete data set and the quality checked data set. Various statistical and heart rate based features were extracted from both data sets. Six machine learning algorithms were trained, compared and evaluated using the selected features. The machine learning algorithms used in this thesis were logistic regression, decision tree, random forest, kNN, SVM and Naive Bayes.

The performance of the algorithms was evaluated using the following metrics: accuracy, precision, recall, specificity, F1-score and mean absolute error. A confusion matrix for each algorithm was presented to visualize the performance. The best performance was obtained with a random forest classifier using the complete data set with an overall accuracy of 81 percent. Sleep was detected with 86 percent accuracy, and wakefulness was detected with the accuracy of 74 percent. The algorithms trained with the quality checked data set did not perform well. The specificity metric unveils that the algorithms were not able to detect the wakefulness state.

The comparison between the two data sets provided valuable information. It was discovered that the distorted signal helped to separate wakefulness and asleep states. Thus,

the overall performance of the method was improved by using both good and bad quality data. It was suggested that the good quality signal was too similar for both the asleep and awake classes which made it more difficult to separate the classes.

Wearable-based sleep detection using PPG data provides a good option for remote health monitoring for the public users. In the future, the level of sleep using PPG data could be studied. The separation of various sleep stages would provide more insights to the quality of the sleep.

References

[1] H. M. Sajjad Hossain, N. Roy, and M. A. Al Hafiz Khan, "Sleep well: A sound sleep monitoring framework for community scaling", in *2015 16th IEEE International Conference on Mobile Data Management*, vol. 1, 2015, pp. 44–53. DOI: 10.1109/MDM.2015.42.

[2] J. Pietilä, E. Helander, T. Myllymäki, I. Korhonen, H. Jimison, and M. Pavel, "Exploratory analysis of associations between individual lifestyles and heart rate variability -based recovery during sleep", in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 2339–2342. DOI: 10.1109/EMBC.2015.7318862.

[3] Y. Zhang, M. Tsujikawa, and Y. Onishi, "Sleep/wake classification via remote ppg signals", in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 3226–3230. DOI: 10.1109/EMBC.2019.8857097.

[4] M. A. Motin, C. Kumar Karmakar, T. Penzel, and M. Palaniswami, "Sleep-wake classification using statistical features extracted from photoplethysmographic signals", in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 5564–5567. DOI: 10.1109/EMBC.2019.8857761.

[5] S. Im, H. Kim, C. Lee, and H. Oh, "Implement of sleep monitoring system using uwb, ppg", in *2019 International Conference on Information and Communication*

Technology Convergence (ICTC), 2019, pp. 1390–1393. DOI: 10.1109/ICTC46691.2019.8939792.

[6] S. Eyal and A. Baharav, "Sleep insights from the finger tip: How photoplethysmography can help quantify sleep", in *2017 Computing in Cardiology (CinC)*, 2017, pp. 1–4. DOI: 10.22489/CinC.2017.274-197.

[7] V. Gupta, S. Mittal, S. Bhaumik, and R. Roy, "Assisting humans to achieve optimal sleep by changing ambient temperature", in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2016, pp. 841–845. DOI: 10.1109/BIBM.2016.7822635.

[8] A. Iwasaki, C. Nakayama, K. Fujiwara, Y. Sumi, M. Matsuo, M. Kano, and H. Kadotani, "Development of a sleep apnea detection algorithm using long short-term memory and heart rate variability", in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 3964–3967. DOI: 10.1109/EMBC.2019.8856463.

[9] A. Sadeh, P. J. Hauri, D. F. Kripke, and P. Lavie, "The role of actigraphy in the evaluation of sleep disorders", in *Sleep*, vol. 18, 1995, pp. 288–302. DOI: 10.1093/sleep/18.4.288.

[10] *Impact of actigraphy in clinical research*, https://www.appliedclinicaltrialsonline.com/view/impact-actigraphy-clinical-research, Accessed:23-April-2021.

[11] A. Sadeh, "The role and validity of actigraphy in sleep medicine: An update", 2011. DOI: 10.1016/j.smrv.2010.10.001.

[12] *Wearable device sales have jumped more than 30 percent this year, exec says*, https://www.cnbc.com/2020/11/20/samsung-wearable-device-sales-are-up-more-than-30percent-this-year.html, Accessed:25-April-2021.

[13] M. Radha, P. Fonseca, A. Moreau, M. Ross, A. Cerny, P. Anderer, X. Long, and R. Aarts, "Sleep stage classification from heart-rate variability using long short-term memory neural networks", 2019. DOI: 10.1038/s41598-019-49703-y.

[14] *Using reflectometry for a ppg waveform*, https://pdfserv.maximintegrated.com/en/an/AN6547.pdf, Accessed:12-March-2021.

[15] K. Budidha and P. A. Kyriacou, "Investigation of photoplethysmography and arterial blood oxygen saturation from the ear-canal and the finger under conditions of artificially induced hypothermia", in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2015, pp. 7954–7957. DOI: 10.1109/EMBC.2015.7320237.

[16] G. Chan, R. Cooper, M. Hosanee, K. Welykholowa, P. Kyriacou, D. Zheng, J. Allen, D. Abbott, N. Lovell, R. Fletcher, and M. Elgendi, "Multi-site photoplethysmography technology for blood pressure assessment: Challenges and recommendations", in *Journal of Clinical Medicine*, vol. 8, 2019. DOI: 10.3390/jcm8111827.

[17] S. Sangurmath and N. Daimiwal, "Application of photoplethysmography in blood flow measurement", in *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, 2015, pp. 929–933. DOI: 10.1109/IIC.2015.7150877.

[18] N. de Pinho Ferreira, C. Gehin, and B. Massot, "Ambient light contribution as a reference for motion artefacts reduction in photoplethysmography", in *13th International Conference on Biomedical Electronics and Devices*, 2020, pp. 1–4. DOI: 10.5220/0008878800230032.

[19] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H. T. Lin, in *Learning from data*. AMLBook, 2012, vol. 4.

[20] *Logistic regression*, https://scikit-learn.org/stable/modules/linear_model.html, Accessed:10-April-2021.

[21] *Logistic regression pros cons*, https://holypython.com/log-reg/logistic-regression-pros-cons/, Accessed:11-April-2021.

[22] S. Ray, *Commonly used machine learning algorithms (with python and r codes)*, https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/, Accessed:3-April-2021.

[23] *A complete guide to the random forest algorithm*, https://builtin.com/data-science/random-forest-algorithm, Accessed:10-April-2021.

[24] A. Statnikov, C. F. Aliferis, D. P. Hardin, and I. Guyon, in *A Gentle Introduction to Support Vector Machines in Biomedicine*. 2011, vol. 1.

[25] *Introduction to k-nearest neighbors: A powerful machine learning algorithm (with implementation in python r)*, https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/, Accessed:10-April-2021.

[26] H. Rajaguru and S. K. Prabhakar, in *KNN Classifier and K-Means Clustering for Robust Classification of Epilepsy from EEG Signals*. Anchor Academic Publishing, 2017.

[27] *Naive bayes from scratch*, https://courses.analyticsvidhya.com/courses/naive-bayes?utm_source=blog&utm_medium=common-machine-learning-algorithms, Accessed:11-April-2021.

[28] A. Mishra, *Metrics to evaluate your machine learning algorithm*, https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234, Accessed:3-April-2021.

[29] D. E. Zomahoun, "A semantic collaborative clustering approach based on confusion matrix", in *2019 15th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, 2019, pp. 688–692. DOI: 10.1109/SITIS.2019.00112.

[30] K. Nighania, *Various ways to evaluate a machine learning model's performance*, https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15, Accessed:18-April-2021.

[31] E. Burns, *Data preparation*, https://searchbusinessanalytics.techtarget.com/definition/data-preparation, Accessed:4-April-2021.

[32] *The data analysis process: 5 steps to better decision making*, https://www.bigskyassociates.com/blog/bid/372186/The-Data-Analysis-Process-5-Steps-To-Better-Decision-Making, Accessed:22-April-2021.

[33] M. Asgari Mehrabadi, I. Azimi, F. Sarhaddi, H. Niela-Vilén, S. Myllyntausta, S. Stenholm, N. Dutt, P. Liljeberg, and A. A. M.Rahmani, "Sleep tracking of a commercially available smart ring and smartwatch against medical-grade actigraphy in everyday settings: Instrument validation study", in *JMIR Mhealth Uhealth*, 2020. DOI: 10.2196/20465.

[34] *Samsung gear sport*, https://www.samsung.com/global/galaxy/gear-sport/, Accessed:09-May-2021.

[35] *Actigraph*, https://actigraphcorp.com/actigraph-wgt3x-bt/, Accessed:09-May-2021.

[36] D. E. McMillan, "Interpreting heart rate variability sleep/wake patterns in cardiac patients", in *The Journal of Cardiovascular Nursing: October 2002*, vol. 17, 2002, pp. 69–81.

[37] P. Stein and Y. Pu, "Heart rate variability, sleep and sleep disorders", in *Sleep Medicine Reviews*, vol. 16, 2012, pp. 47–66. DOI: https://doi.org/10.1016/j.smrv.2011.02.005.

[38] *What is data labeling?*, https://aws.amazon.com/sagemaker/groundtruth/what-is-data-labeling/, Accessed:5-April-2021.

[39] A. Mahmoudzadeh, I. Azimi, A. Rahmani, and P. Liljeberg, "Lightweight photoplethysmography quality assessment for real-time iot-based health monitoring using unsupervised anomaly detection", in *Elsevier International Conference on Ambient Systems, Networks and Technologies (ANT'21)*, 2021.

[40] *Feature extraction*, https://deepai.org/machine-learning-glossary-and-terms/feature-extraction, Accessed:11-April-2021.

[41] P. van Gent, H. Farah, N. Nes, and B. Arem, "Heart rate analysis for human factors: Development and validation of an open source toolkit for noisy naturalistic heart rate data", Jun. 2018.

[42] P. van Gent, B. Arem, H. Farah, and N. Nes, "Analysing noisy driver physiology real-time using off-the-shelf sensors: Heart rate analysis software from the taking the fast lane project", Nov. 2018. DOI: 10.13140/RG.2.2.24895.56485.

[43] *Feature selection techniques in machine learning with python*, https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e, Accessed:11-April-2021.

[44] S. Nembrini, I. R. König, and M. N. Wright, "The revival of the gini importance?", in *Bioinformatics*, vol. 34, 2018, pp. 3711–3718. DOI: 10.1093/bioinformatics/bty373.

[45] A. Dubey, *Feature selection using random forest*, https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f, Accessed:28-March-2021.

[46] *How to handle imbalanced classes in machine learning*, https://elitedatascience.com/imbalanced-classes, Accessed:18-April-2021.

[47] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification", in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 2016, pp. 78–83. DOI: 10.1109/IACC.2016.25.

[48] *Preprocessing data*, https://scikit-learn.org/stable/modules/preprocessing.html, Accessed:3-April-2021.

[49] *Scikit learn*, https://scikit-learn.org/, Accessed:09-May-2021.