# Epileptic seizure detection using near-ear EEG

BSc Thesis

Chris Bot, Arthur de Groot & Aurore de Spirlet

TU Delft
Delft University of Technology

TU Delft

# Epileptic seizure detection using near-ear EEG

*A Brain-Computer Interface Inside Your Earphones*

by

Chris Bot, Arthur de Groot and Aurore de Spirlet

Bachelor Graduation Thesis


| | | |
|---|---|---|
| Student Number: | C. Bot | 4927133 |
| | A. de Groot | 5068916 |
| | A. de Spirlet | 4876229 |
| Project Duration: | April 19, 2022 - June 24, 2022 | |
| Supervisor: | dr. T. Costa | TU Delft |
| | dr. D. Muratore | TU Delft |
| Daily Supervisor: | BSc. P. Burgar | TU Delft |

DELFT UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRICAL ENGINEERING, MATHEMATICS AND COMPUTER SCIENCE

ELECTRICAL ENGINEERING PROGRAMME

# Abstract

60 million people around the world have epilepsy, which is a neurological disorder that severely impacts their day to day life negatively. Currently available methods to reduce the effects of epilepsy are either ineffective or require expensive and invasive surgery. A new method has been found that can suppress epilepsy without the need of surgery, called Transcutaneous Vagus Nerve Stimulation (t-VNS). Detecting epileptic seizures is important for this method, as the stimulation should only be used during a seizure. Traditionally, detecting epilepsy is done using scalp-Electroencephalography (EEG), which requires a controlled environment and is hard to use in day to day life. Recently, advancements have been made in ear-EEG, which allows for EEG outside a controlled environment. This study focuses on detecting epilepsy using ear-EEG. Ear-EEG was simulated using scalp-EEG channels close to the ear. After low-pass filtering and downsampling the results were obtained using features obtained from the Wavelet Transform (WT) and Fourier Transform (FT) in combination with several Machine Learning (ML) models; these being a random forest, a Support Vector Machine (SVM), and a Neural Network (NN). Furthermore PCA was also applied to the features, with a threshold of 0%, 95% and 99%. The results clearly show that using the WT outperforms features from the FT. Furthermore, out of the three models, the NN consistently has the best sensitivity for detecting seizures. The best sensitivity was achieved using WT features with a NN and a threshold of 99% for the PCA. The accuracy and sensitivity are 99.3% and 83.5% respectively, which is comparable to previous ear-EEG based research detecting epileptic seizures.

# Preface and Acknowledgments

This report was written in the fourth quarter of academic year 2021/2022 in the context of the Bachelor Graduation Project to obtain the Electrical Engineering Bachelor degree at the TU Delft.
The project was conducted with a team of 12 students under the supervision of dr. Dante Muratore and dr. Tiago Costa from the Bio-electronics department. The group was split in teams of three people such that each team worked on a different part of the complete system, namely an in-ear EEG device to detect and suppress epileptic seizures.
The past weeks have not only let us integrate and apply all the knowledge acquired through our Bachelor but have also deepened our knowledge in certain areas such as Signal Processing and Machine Learning. Important skills such as reading and identifying relevant literature sources related to our project have also been obtained. We really enjoyed having the liberty to make our own decision to obtain a final successful working system, despite the short amount of time. Most importantly, this project has let us realize how engineers, like us, can make serious progress towards majorly improving the life of people with previously incurable diseases.
The cooperation within our subgroup went really well. We saw each other three times a week to facilitate communication and discussed improvements and eventual new ideas of implementation. All of us were very enthusiastic and motivated about the project and we all have learnt a lot. Not only have we sharpened our skills as electrical engineers, but we have also learnt about EEG's and the current situation of treatment for people affected by epilepsy.
Finally, we would like to thank dr. Dante Muratore and dr. Tiago Costa, our supervisors, for their time, effort and knowledge they shared with us during the meetings and in their free time. We are also thankful for them accepting to supervise two teams of 6 people that were all evenly motivated to start this project. Another thank you to dr. Bori Hunyadi, who previously worked on patient-specific seizure monitoring, for kindly answering our questions.

# Contents


**1 Introduction** **5**
- 1.1 Problem definition . . . 5
- 1.2 Goal of the Project . . . 6
- 1.3 Thesis outline . . . 6
- 1.4 Background Information . . . 7
  - 1.4.1 Electroencephalography . . . 7
  - 1.4.2 Epilepsy . . . 7

**2 Programme of Requirements** **8**
- 2.1 Functional requirements . . . 8

**3 Pre-processing** **9**
- 3.1 Data source . . . 9
- 3.2 Channel selection . . . 10
- 3.3 Artefacts and noise sources . . . 10
- 3.4 Filtering . . . 11
- 3.5 ASR . . . 13
- 3.6 Downsampling . . . 14
- 3.7 Epoching . . . 14
- 3.8 Conclusions . . . 14

**4 Feature Extraction** **15**
- 4.1 Frequency Domain Analysis . . . 15
  - 4.1.1 Fourier Analysis . . . 15
  - 4.1.2 Power Spectrum . . . 16
  - 4.1.3 Windowing . . . 17
  - 4.1.4 Feature Extraction . . . 17
- 4.2 Time-Frequency Domain Analysis . . . 18
  - 4.2.1 Continous Wavelet Transform . . . 18
  - 4.2.2 Discrete Wavelet Transform (DWT) . . . 18
  - 4.2.3 Feature Extraction . . . 19
  - 4.2.4 Normalization . . . 20
- 4.3 Dimensionality Reduction . . . 20

**5 Classification** **22**
- 5.1 Machine Learning . . . 22
  - 5.1.1 Hyperparameter Tuning . . . 22
- 5.2 Data Preparation . . . 23
- 5.3 Machine Learning Models . . . 23
  - 5.3.1 Random Forest . . . 23
  - 5.3.2 Support Vector Machine . . . 23
  - 5.3.3 Neural Networks . . . 24

**6 Discussion of Results** **26**
6.1 Results . . . 26
6.1.1 Analysed data . . . 26
6.1.2 Epoch Length . . . 26
6.1.3 Hyperparameters . . . 27
6.2 Results Per Patient . . . 27
6.3 Results of Grouped Patients . . . 28
6.4 Processing time . . . 28
6.5 Discussion . . . 28

**7 Conclusion** **29**
7.1 Future Work . . . 29

**List of abbreviations** **31**

**Bibliography** **32**

**A Additional Figures & Tables** **37**
A.1 Background Information . . . 37
A.2 Pre-Processing . . . 38
A.3 Feature Extraction . . . 38
A.3.1 Overlapping windows . . . 38
A.3.2 Feature dimensions . . . 39
A.4 Classification . . . 39
A.4.1 SVM . . . 39
A.5 Discussion of Results . . . 39
A.5.1 All Results . . . 39

**B Training results** **44**
B.1 Hyperparameter tuning results . . . 44
B.2 Results per patient . . . 44
B.2.1 Random Forests . . . 44
B.2.2 Support Vector Machine . . . 45
B.2.3 Neural Network . . . 46

# Chapter 1

# Introduction

## 1.1 Problem definition

Epilepsy is a chronic neurological disease that affects 60 million people worldwide, making it the most common of its type [1]. The disease is characterised by recurrent seizures that occur at unpredictable moments in time. These seizures result in unconscious convulsions of the body and because of this, people suffering from epilepsy must be constantly aware of their surroundings. Seizures in unfortunate circumstances could result in sudden fractures, drownings and burns. Epilepsy causes a major burden for people who want to live a normal, peaceful life.
It is however not impossible to treat; one option is to prescribe anti-epileptic drugs to patients, but these medications have side effects, such as drowsiness, lack of energy and agitation [2]. Also, about a third of people with epilepsy are drug resistant [3]. Another option is to surgically implant a device that can do Vagus Nerve Stimulation (VNS) in order to suppress or even stop epileptic seizures. However, this method is underutilised due to the high cost, tendency to be afraid of the surgery, and lack of trust in the device [4]. A new methodology in 2015 has been developed though, which is called Transcutaneous Vagus Nerve Stimulation (t-VNS) [5]. This new technology is non invasive and stimulates the vagus nerve through electrodes on the skin. This method should produce therapeutic effects that are similar to those produced by regular vagus nerve stimulation.

Using Electroencephalography (EEG) it is possible to measure and record electrical activity of the brain. During a seizure these recorded brain waves are disturbed and this can be detected, meaning that EEG can be used to detect epileptic seizures. When both t-VNS and EEG are combined, a closed loop feedback system can be created; detecting and preventing epilepsy then becomes possible in a non-invasive way. While it is possible to detect upcoming or ongoing epileptic seizures with EEG, it requires placement of electrodes all over the scalp. This is very invasive and unpractical for everyday life. Therefore, a new approach to monitoring brain activity has gotten some interest in the past few years. This approach, namely in-ear EEG or ear-EEG, consists of placing electrodes in (and around) the ear instead of on the scalp. Several advantages can be observed compared to scalp-EEG, such as diminished motion artifacts, reduced interference from external electric fields due to the ear canal being enclosed by an electrical conductive medium, and finally it is discreet [4]. Challenges come along with this new setup such as the decreased number of electrodes and the limited power available for recording and detecting epileptic seizures through machine learning.

Very few studies report successful seizure detection using electrodes placed inside or behind the ear and no such EEG setup that is small and unobtrusive enough to be used in everyday life exists yet [6].
The focus of this thesis is to create a detection system that uses EEG as an input to determine whether this input EEG data contains a seizure or not. The pre-processing steps applied on the data, the type of features to be extracted from the data, and finally, the type of classification techniques or algorithms to be applied on these extracted features to detect the label, are considerations to be taken into account.

## 1.2 Goal of the Project

The larger scope of the project aims to develop a wearable device that not only detects epileptic seizures using in-ear EEG but also stimulates the vagus nerve using an electric circuit and electrodes. This means that the full device is able to detect and suppress seizures by itself; making it a closed feedback-loop system. Some requirements will have to be met in order to create such an ear-EEG based closed feedback-loop system. Firstly, a circuit has to be designed that can safely stimulate the vagus nerve. Safety and low power usage are a major concern as the device will need to be able to work the entire day.
Secondly, an artificial ear has to be made that can be used to test the ear-EEG device and the aforementioned stimulus circuit. It is extremely important to be able to test these devices for safety reasons, as they should not damage the skin and hurt the user.
Thirdly, the measured EEG signal has to be processed. There are two ways in which this is done: detecting epileptic seizures and authenticating the user of the device. The importance of the former has already been discussed and as each person has unique EEG signals [7] it is important to properly authenticate the user. The device's t-VNS output current has to be tuned per person; a wrongly tuned device could carry some safety hazards, such as burned skin tissue.

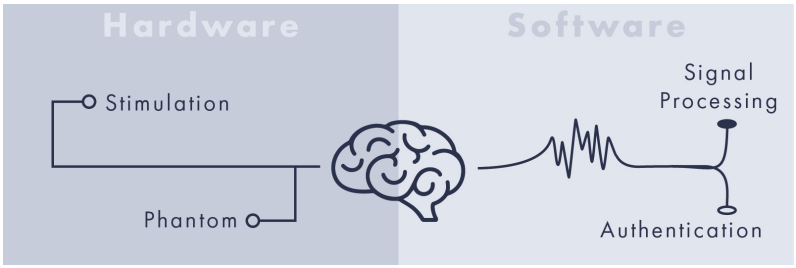



Figure 1.1: Project overview and subgroups.

The project members are divided into four subgroups that each tackle one of the aforementioned problems. These can be seen in Figure 1.1 and include the following groups:

- Stimulation: create a circuit that can safely perform t-VNS.
- Phantom: create an artificial ear that can be used for testing.
- Signal Processing: create a system that can detect epileptic seizures and control the stimulus circuit.
- Authentication: create a system that can use EEG recordings to authenticate the user.

## 1.3 Thesis outline

This thesis concerns the signal processing subgroup that tackles the process of detecting seizures from raw digital EEG recordings. Figure 1.2 shows the plan of approach. The entire system is divided into three modules: raw EEG data pre-processing, feature extraction and selection, and classification.
First, chapter 2 covers the requirements of the detection system. In chapter 3 the used data-set and electrode as well as what kind of filtering is applied to the raw EEG data in order to suppress any unwanted noise and artifacts are discussed. In chapter 4 methods of feature extraction from the pre-processed data are implemented to obtain useful and discriminate characteristics. Chapter 5 discusses how these features are

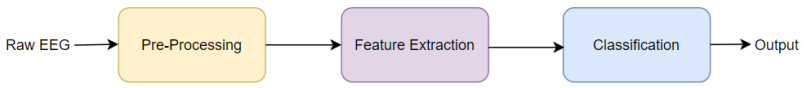



Figure 1.2: Signal processing pipeline

used in order to train a Machine Learning (ML) model to classify epileptic seizures. The results obtained from these different extraction methods and algorithms are analysed and discussed in chapter 6. The remaining sections of this chapter discuss some important background information.

## 1.4 Background Information

### 1.4.1 Electroencephalography

Before one can think about processing signals, one should know and understand the signals in question. As mentioned before, electroencephalography is a method that can be used to record brain activity. What is actually being measured is the activity of millions of cortical neurons that produce an electric field [8]. Potential differences in this electric field can be measured using multiple electrodes; this measurement is called an electroencephalogram. EEG recordings have frequency content ranging from 0.01 Hz to 100 Hz and their amplitude vary from a few microvolts to approximately 100 $\mu V$, according to [8]. Furthermore, EEG signals can be seen as stochastic, as their characteristics cannot be predicted. EEG is done using wet or dry electrodes placed on the scalp to track the different brain waves. These waves can be distinguished into five different frequency ranges depicted in Table A.1 [9].

The identification and tracking of these frequency ranges is mainly used for medical purposes, for example detecting sleep quality and monitoring depression and/or anxiety. While brain monitoring through the scalp gives the possibility to record large amounts of signals from every part of the cerebral cortex, it can be invasive and unpractical. Furthermore, this monitoring usually requires the patient to be in a medical facility for extended periods of time. Therefore, a new approach to monitor brain activity has gotten some interest in the past few years. This approach, namely ear-EEG, consists of placing electrodes inside and around the ear instead of on the scalp. This new method leads the path to the creation of discrete wearables to monitor the health of people in an easy way. To name a few, a study provides a proof of concept for successful in-ear monitoring of drowsiness in patients [10], another proposes a wearable in-ear for overnight sleep monitoring [11] and finally [12] confirms the possibility of classifying attention states with in-ear EEG. Using ear-EEG to detect seizures however is still a field that requires a lot of research to obtain a proof of concept.

#### Ear-EEG device prototype

The systems developed in this thesis are meant to be used in an ear-EEG device created by Patricija Burgar and Cyril Weustink. While development on it has not yet finished, a prototype for the device can be seen in Figure A.2. It cannot be used for measuring ear-EEG signals for this project in its unfinished state. Furthermore, even if the device were finished, the procedures required to measure epileptic patients is much longer than the duration of this project. Hence, in chapter 3, alternative methods for data acquisition will be discussed.

#### EEGLAB

EEGLAB is an interactive toolbox made for use with MATLAB [13]. The toolbox allows for processing and visualisation of EEG data. It has been used in this thesis to import EEG data from data files and perform most of the steps taken in pre-processing.

### 1.4.2 Epilepsy

Epilepsy is a neurological disorder that affects the normal activity of your brain and thus results in unusual behaviors or seizures. The epileptic seizure process can be divided into four phases: ictal, preictal, interictal and postictal. The ictal phase represents the moment when the seizure is happening. The preictal phase happens right before the ictal or outbreak phase. The Interictal phase is the instance between two consecutive seizures and, finally, the postical phase is the phase after a seizure attack.

# Chapter 2

# Programme of Requirements

The goal of this thesis is to present a signal processing system that can detect epilepsy using ear-EEG data. The result of detection is then used to control a circuit that can use t-VNS to suppress seizures. This means that the signal processing system needs a high accuracy for detecting epileptic seizures (high sensitivity). Also, this device is to be worn all day, meaning that low power usage is preferred. The system should thus also not enable the stimulus device unnecessarily, as to not waste energy. However, a high accuracy means a more complex model, which requires more power. In the end, the decision was made to choose to improve accuracy over improving power efficiency. This is because a missed epileptic seizure could be devastating.

## 2.1 Functional requirements

- The system must use only EEG data to detect epileptic seizures.
- In case of scalp-EEG, the system may only use channels placed close to the ears.
- The system must be able to detect epilepsy in real time; EEG recordings must be recorded and processed at least every 6 seconds.
- With the aforementioned limitations, the system must have a sensitivity of at least 95% and an accuracy of at least 99% on these short recordings.
- The system must be able to classify seizures in less time than the recording time.
- The system must be able to control the stimulus circuit.
- A hardware implementation of the system should be possible.
- Computational complexity and memory storage size should be as low as possible in a hardware implementation.

The accuracy is calculated as follows $\frac{TP+TN}{TP+FP+TN+FN} * 100\%$ and sensitivity through the following equation $\frac{TP}{TP+FN} * 100$. Where $TP$ is the number of correct seizure classifications, $FP$ is the number of non seizures being incorrectly classified as seizures, $TN$ is the number of non seizures being classified correctly, and $FN$ is the number of seizures being incorrectly classified as normal EEG activity. The reason for the recording time is that epileptic seizures last about 30 to 120 seconds [14]. Using the required sensitivity of 95%, the chance of not detecting a seizure before it is over in the worst case is $0.05^{30/6} = 0.00003125\%$, which is acceptably low.

# Chapter 3

# Pre-processing

The raw, unfiltered signal measured by the ear-EEG device cannot be used directly. This is because much of the important information is mixed with or distorted by unwanted signal components. One of these components is noise, with sources such as the noise of used electronics [15] and $1/f$ noise (or pink noise) [8]. Artefacts are also, by definition, included in these unwanted signal components. There are numerous sources of artefacts in an EEG, apart from noise, including but not limited to Electromyography (EMG) and Electrocardiography (ECG).
In the coming sections the artefacts and noise in the signal and the methods that are used to (partially) remove them through various means will be explored.
Important to mention is that no ear-EEG device and subjects with epilepsy were available, which means that existing data sets were used. As there are no public datasets for epilepsy patients with ear-EEG recordings, datasets of epileptic patiens with scalp-EEG were used. The used data set contains scalp-EEG instead of ear-EEG, which means that some steps must be taken in order to approximate ear-EEG. Also, these two signals have some differing properties, meaning different choices have to be made during pre-processing.

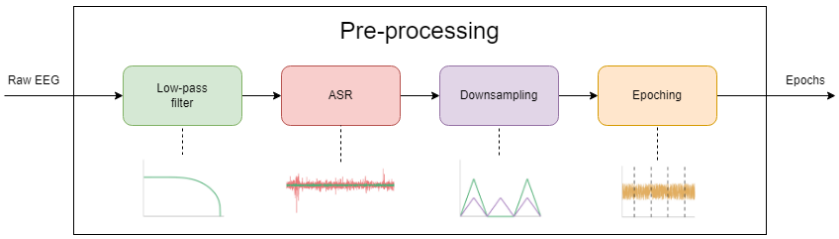



Figure 3.1: Pre-processing pipeline

Figure 3.1 shows the full pre-processing pipeline. All the steps shown in this pipeline will be discussed in more detail in the coming sections.

## 3.1 Data source

The EEG data used in this study is the data from the CHB-MIT scalp-EEG Database [16, 17].
This data set consists of 24 young patients aged between 1.5 – 19 years old who experience seizures. Every recording is at least one hour of digitised EEG signals, although some are 2 to 4 hours in length. The measurements were done using the *10-20 system* [18, 19] with a sampling rate of 256 Hz. The 10-20 system is an internationally recognised system that describes the placement of electrodes around the scalp and is visualised in Figure 3.2. The recordings contain 23 channels of scalp-EEGs. In total the data set contains 664 recordings of which 129 contain one or more seizures. In the 129 recordings, there is a total of 198 recorded seizures. The used channels and timestamp of the seizures per recording are noted in a

summary file that is provided with the data set. Every step of pre-processing discussed after this is done on per hour-long recording file from this data set.

## 3.2 Channel selection

The goal of this project is to find out if ear-EEG can be used to detect epileptic seizures. As mentioned before, a scalp-EEG data set is used, considering no appropriate ear-EEG data set is available. No other research has attempted to simulate ear-EEG in this way.

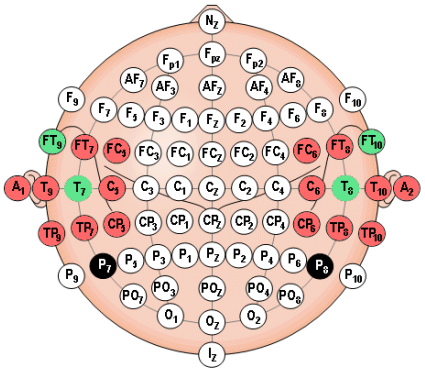



Figure 3.2: 10-20 system, electrodes in red are selected to simulate ear-EEG, electrodes in green were actually available [16].

To simulate ear-EEG, only the channels that are close to the ears are used. The electrodes in red shown in Figure 3.2 are the ones that are selected from the recordings in order to attempt to simulate ear-EEG. Furthermore, of the 20 selected channels, only four are available in the CHB-MIT data set. In total there are six channels though, due to the T7 and T8 electrodes being used twice with different reference electrodes.

## 3.3 Artefacts and noise sources

As mentioned before, there are quite a few artefacts and noise sources that can obscure or distort useful information from a signal. In the case of EEG a distinction can be made between physiological artefacts and non-physiological artefacts. The former is a result of physiological effects, which are effects stemming from functions of the body. These include ECG artefacts – the heartbeat – and EMG artefacts that come from the movement of muscles. In ear-EEG the most apparent measurable EMG artefacts stem from jaw movement, eyelid movement, breathing and speech [20, 21]. It is especially important to remove EMG artefacts for the application of everyday use, as these have a high amplitude. The body moves a lot during a day and that causes a lot of these artefacts, which could influence the final classification result.
Non-physiological artefacts are generated by sources outside of the body. Some examples are the Alternating Current (AC) frequency of the power supply (50 Hz in Europe), or the electrode not being placed correctly [22]. Another source could be malfunctioning equipment, which can add extra noise to the EEG signal. However, as mentioned before, the CHB-MIT data set is used. This means that the source of non-physiological artefacts that exist in the used data are uncertain.

In order to find ways to reduce or remove the effects of these artefacts on the signal, it is useful to study some of their properties. ECG artefacts have a relatively constant frequency in the range of 60 to 100 beats per minute (bpm) [23]. In the frequency domain this is visible as an energy peak around 1 to 1.7 Hz. EMG artefacts are caused by movement of the body and cause big spikes in measured EEG data; these artefacts have a high variance and only happen when the body moves [24]. In Figure 3.3 EMG artefacts are especially visible in the part of the data showing a seizure, as the patient is then moving.
As was previously stated, the non-physiological artefacts are harder to deal with, since their sources are

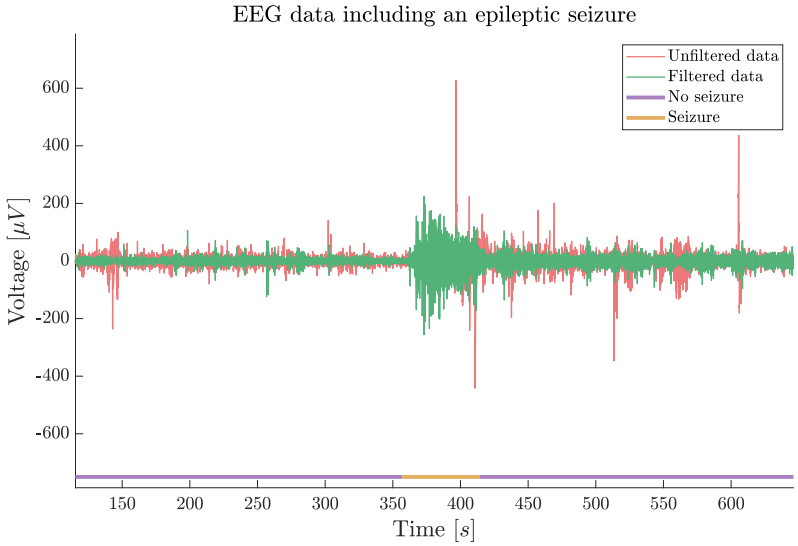



Figure 3.3: Segment of EEG data from a single channel that shows a seizure. The filter used is a low-pass sinc Finite Impulse Response (FIR) filter with a cutoff at 30 Hz.

mostly the used electronics [15]. An exception to this is the noise observed due to the power grid to which the measuring equipment is connected. In Europe the AC grid oscillates at 50 Hz, while in the US this value is 60 Hz. As the used data set was created in the US we expect to find this artefact at 60 Hz and it is visible at that frequency in Figure 3.4. In Figure 3.3 one can see that after the seizure the channel is a lot noisier, which could be the result of one of the electrodes being moved during the seizure, an example of a non-physiological artefact. This information is important when choosing specifications of pre-processing methods later on.

Noise sources that occur in ear-EEG include $1/f$ noise (pink noise) and noise generated by the use of electrical components, such as thermal noise or shot noise [15]. The former, pink noise, can be partially negated by certain circuit methods such as Correlated Double Sampling (CDS) [25]. This is something that is done by an analog circuit while measuring, which means that we are reliant on the CHB-MIT data set to have applied these methods.

As can be seen in the Power Spectral Density (PSD) shown in Figure 3.4, $1/f$ noise has been negated in the CHB-MIT data set. This can be seen due to there being no correlation between the (un)filtered data and the pink noise at lower frequencies. There is also no distinguishable peak around 1 to 1.7 Hz, which means that ECG artefacts are most likely too insignificant to affect the final classification result.

Electronic noise is a problem that can most easily be dealt with by selecting proper specifications for the electronics used during measuring and by analog noise reducing techniques, similarly to $1/f$ noise [15]. Again, this is reliant on the measurement setup used by the providers of the CHB-MIT data sets.

## 3.4 Filtering

A filter is a device that can be used to get the desired components from a given object. In signal processing filters are used to separate a signal from unwanted components such as noise or extract specific frequency bands. This is very important during pre-processing as there is a limited frequency band in which useful information is contained. Frequency bands that are useful for detecting epilepsy lie in the $\delta$ to low $\gamma$ bands [17] as they have been defined in Figure A.1. To be more precise, the more important frequencies range from 0.5 Hz to about 30 Hz. Based on existing literature and available tools a sinc FIR filter was chosen [26, 13].

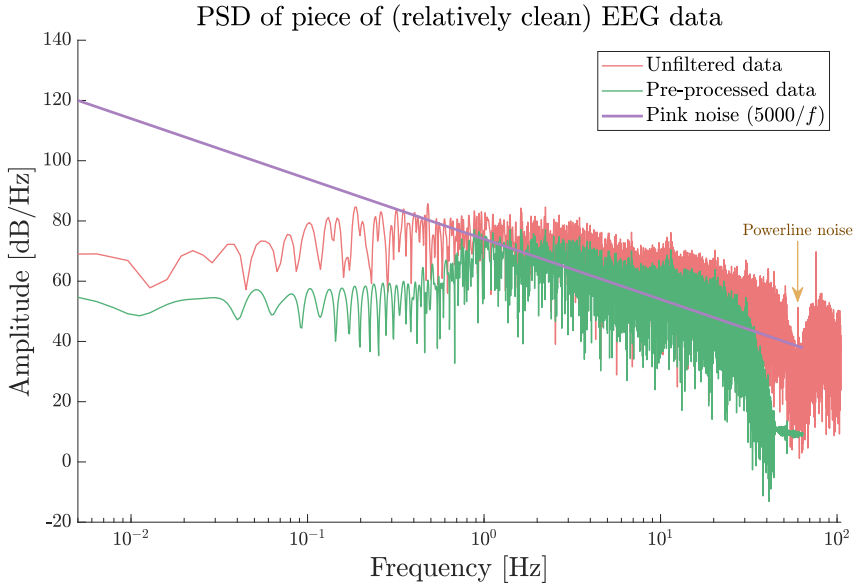



Figure 3.4: PSD of small quiet section of a single channel of (un)filtered data and $1/f$ noise. The constant of 5000 has been chosen to line up the amplitudes of the signal and noise.

Ideally the filter must have cutoffs (-6 dB) at 0.5 Hz and 30 Hz. In real life though the low cutoff frequency means that the entire delta band is attenuated when using realistically achievable filter orders. This is problematic since the delta band contains useful information for classifying seizures. A low pass filter with a cutoff at 30 Hz can also be used. The advantage of this is that the delta band is not attenuated, however, artefacts and Direct Current (DC) components of the signal are not filtered out.
The order of the filter determines its roll-off. This means that the higher the order, the less unwanted signal components remain after filtering. A higher filter order also means more computational complexity, so a trade-off has to be made here. As the computation must be lower than the time of recording, an order of 30 was chosen. This gives a computation time of about 0.2 seconds to filter all selected channels, each containing an hour's worth of EEG data.

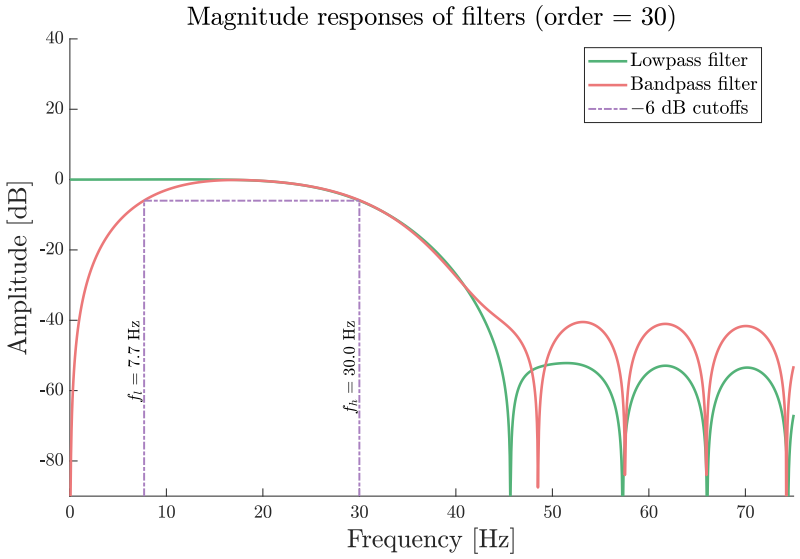



Figure 3.5: Magnitude response of low-pass and band-pass filter. Both filters have a high cutoff ($f_h$) at 30 Hz and the Band-pass has a lower cutoff ($f_l$) at 7.7 Hz.

Figure 3.5 shows the magnitude response of both the low-pass and band-pass filter. The filter was designed to have a lower cutoff at 0.5 Hz, but in the figure one can see that $f_l = 7.7$ Hz. This is because the roll-off of the filter is not steep enough to reach a minimum amplitude at 0 Hz from 0.5 Hz. The tool used to design the filter calculated 7.7 Hz to be the minimum $f_l$ possible with the chosen filter order. For higher filter orders a lower cutoff frequency can be used due to a steeper roll-off; a filter order of 100 gives $f_l = 2.4$ Hz, at the cost of a higher computation time.
All in all, the low-pass filter is the better choice as the $\delta$ to $\theta$ bands could contain very useful information for classifying seizures correctly. This decision does risk low frequency and DC noise, however the most significant low frequency noise, $1/f$ noise, is already negated.
Regardless of the choice, high frequency artefacts and noise will be filtered out. This includes the power grid noise for example.

## 3.5 ASR

Artefact Subspace Reduction (ASR) is a method that can be used to remove high variance components from a signal. [27, 28] contain the rigorous maths showing how exactly the method works. For this study understanding the specific mathematics that make up ASR are not necessary, however it is important to know how the method works. To shortly summarise this, ASR uses a clean artefact-less part of the signal and attempts to reconstruct the rest of the signal using it. This makes it possible to remove high variance components such as artefacts. In real life it is almost impossible to get an artefact-less signal, so instead the least artefact-contaminated part of the signal is selected. A calibration signal could also be recorded in a controlled environment, in order to have an (almost) artefact-less reference signal. The major downside of this method is that it has a high computation complexity, which could potentially make it difficult to implement in hardware.

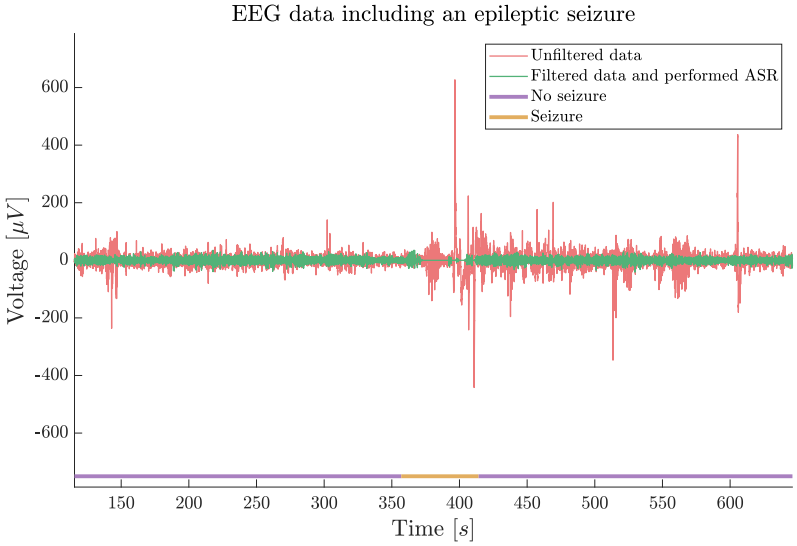



Figure 3.6: Segment of EEG data from a single channel that shows a seizure. Additionally, ASR is applied to the filtered signal.

Figure 3.6 shows the same data from a signal channel as Figure 3.3, but in this case ASR is applied on the signal. One can see the effects best at the part showing a seizure. The EMG artefacts are almost completely removed. It is also notable that the noisier channel after the seizure is now very similar to the data before the seizure.

## 3.6 Downsampling

Downsampling is the combination of using a low-pass or band-pass filter and decimation [29]. The latter is the process of reducing the amount of samples in a signal by a downsampling factor $m \in \mathbb{N}$, which is realised by taking every $m$th sample of the signal. Decimation effectively reduces the sampling frequency of the signal by $m$. The main advantage of downsampling is reducing the size of the data, thus reducing computational cost.
The CHB-MIT data set has all its data sampled at 256 Hz. According to the Nyquist theorem, the highest frequency that can be represented without aliasing is half the sampling frequency [30]. The minimum sampling frequency is 60 Hz as the highest frequency with useful information in the EEG is 30 Hz. The filter is not ideal though, which means that the minimal sampling frequency that can be used while avoiding aliasing is actually around 80 Hz. This means that the smallest sampling frequency that is feasible is 128 Hz, meaning the downsampling factor is 2.

## 3.7 Epoching

The final system has to be able to classify a seizure using a maximum of six seconds of an EEG recording as per the requirements. This is why the data used to train the classifier will also be divided into segments of at most six seconds. Each segment of a few seconds of an EEG is called an epoch.
Choosing the length of each epoch is crucial because it has a direct impact on the frequency resolution [31]. The frequency resolution $f_c$ can be determined through Equation 3.1.

$$f_c = \frac{f_s}{N} \tag{3.1}$$

In this equation $f_s$ represents the sampling frequency and $N$ the number of samples in each epoch.
Hence, with a fixed sampling frequency, the length of an epoch determines the frequency resolution. The bigger the epoch size the better the frequency resolution becomes. However, increasing the epoch size also increases memory space needed for storage and the recording time. An epoch length that is too long could potentially result in missing a seizure. The upper bound is six seconds as stated in the requirements. The lower bound of the epoch length is decided to be three seconds long. The reason why this lower bound is selected is due to the constraint that Welch's method imposes to obtain the PSD of an epoch. This is explained in more detail in subsection 4.1.2.

## 3.8 Conclusions

To summarise, a low-pass sinc FIR filter of order 30 with cutoff at 30 Hz is used, ASR is then performed on the signal to remove high variance artefacts. After this, the signal is downsampled by a factor of two and then divided into epochs of three seconds.
After all the techniques that were discussed are applied to the signal, we are left with six channels of pre-processed data. These can then be used to extract features from in order to train the classification model.
Figure A.3 shows what the data of a few used channels looks like after pre-processing. The figures show a few epochs of four randomly selected channels. The epochs on the left contain no seizure, while those on the right do contain seizures.

# Chapter 4

# Feature Extraction

Feature extraction is necessary to correctly classify whether an individual is having a seizure. This chapter discusses the extraction of effective features from the pre-processed EEG.
EEG's are characterised by their variable amplitude and frequency components and represent different states of brain activity. Therefore, features are extracted from both frequency and time-frequency domain. In subsection 1.4.1 it is explained that EEG's can be decomposed into functionally distinct frequency bands. In order to extract features from these bands, several methods can be used. These include but are not limited to the Fourier Transform (FT) and the Wavelet Transform (WT) [32]. In this study, the FT and WT are applied. The FT can be used to perform spectral analysis on EEG data, which is a commonly used method [33]. Features can then be extracted from the power spectrum of each epoch of an EEG.
The WT addresses the limitations that the FT faces regarding time-frequency resolution. This method extracts features in the time-frequency domain by considering dynamical changes in non-stationary signals. It is known that WT performs better than the FT when analyzing and classifying EEG's, at an expense of computational time [34]. Even so, in this study both the Wavelet and Fourier method will be applied on the EEG's to extract features and later, in chapter 6, the classification performance will be compared for both methods.
This chapter is structured as follows: section 4.1 covers extracting features in the frequency domain by applying the FT, section 4.2 covers time-frequency domain features obtained through the WT and finally feature selection is discussed in section 4.3.

## 4.1 Frequency Domain Analysis

### 4.1.1 Fourier Analysis

The Discrete Fourier Transform (DFT) is applied per epoch of the discrete time EEG. The latter is defined as:

$$X(k) = \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi kn/N} \qquad k = 0, 1, ..., N-1 \tag{4.1}$$

where $x_n$ are the discrete samples of each EEG epoch and $X(k)$ are the amplitudes of the $N$ different samples of the signal.

A faster method is used to transform discrete time domain signals into the frequency domain, namely the

Fast Fourier Transform (FFT). This is an algorithm for fast and efficient computation of the DFT. It reduces the number of computations needed for $N$ samples from $N^2$ to $N \cdot \log_2(N)$.

### 4.1.2 Power Spectrum

From the DFT the PSD can be computed. Several methods can be used to obtain the PSD, for example through the use of periodograms or Welch's method. The periodogram is computed as follows:

$$\hat{P_x}(k) = \frac{|X(k)^2|}{N} \tag{4.2}$$

where $P_x(k)$ is the periodogram, $N$ represents the number of sample in each epoch and $X(k)$ is the DFT of a single epoch.

While this method is simple and straightforward it can cause some issues such as random noise appearing in the PSD. To overcome this issue, Welch's method has been developed. This new method divides the discrete time signal into overlapping segments, computes the periodogram on each segment and averages it out [35]. This can be seen in Equation 4.3. The division of segments is done by windowing that signal in question which is further explained in subsection 4.1.3. In Figure 4.1 the two PSDs can be seen; it is clearly visible that the PSD computed using Welch's method is much less noisy than the simple periodogram.

$$P_{Welch}(k) = \frac{1}{L} \cdot \sum_{m=0}^{L-1} P_{x_m}(k) \tag{4.3}$$

where $L$ is the number of segments and $P_{x_m}$ is the periodogram over one segment.

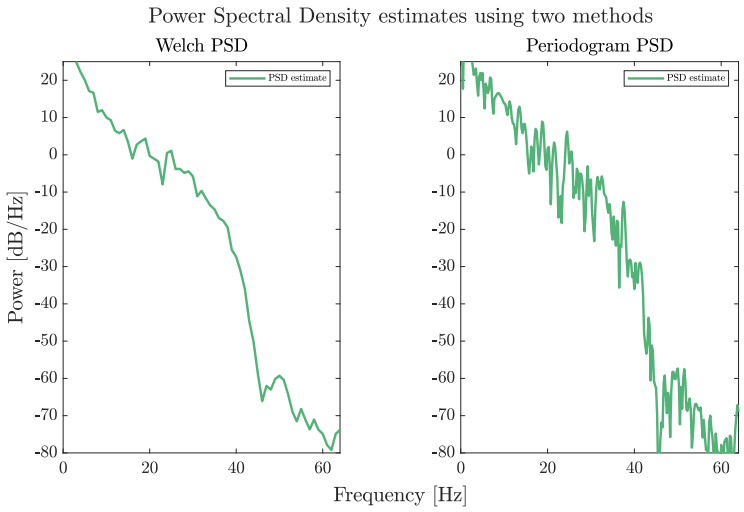



Figure 4.1: Comparison of PSDs of one epoch using Welch's and Periodogram methods.

Welch's method does have some drawbacks though. One of these is a reduction in frequency resolution. This is because each epoch is further divided into segments. The frequency resolution is not dependent anymore on the epoch length but on the length of each segment over which the periodogram is computed. Such as described in Equation 3.1, the frequency resolution is inversely proportional to the number of samples contained in your segment. Therefore since the minimum frequency of interest is 1 Hz each segment must be at least 1 second long. In order to have a reasonable noise-free PSD through Welch's method averaging is performed over 1 second segments.

### 4.1.3 Windowing

Splitting the discrete time epochs into several segments is done by multiplying the signal with a window function. Window functions are characterised by the main and side lobes of their magnitude spectrum. Figure 4.2 shows various types of windows. Choosing the right window is crucial to prevent distortion in the frequency domain of the segmented pieces. Ideally the width of the main lobe is very narrow to increase resolution. The amplitude of the side lobes should be as low as possible to increase the energy on the main lobe and to increase the stop-band attenuation. [35]
A standard method for segmenting a signal is to truncate it, in other words, to use the rectangular window. While this window contains the narrowest main lobe compared to other windows, it also contains the highest side lobes.

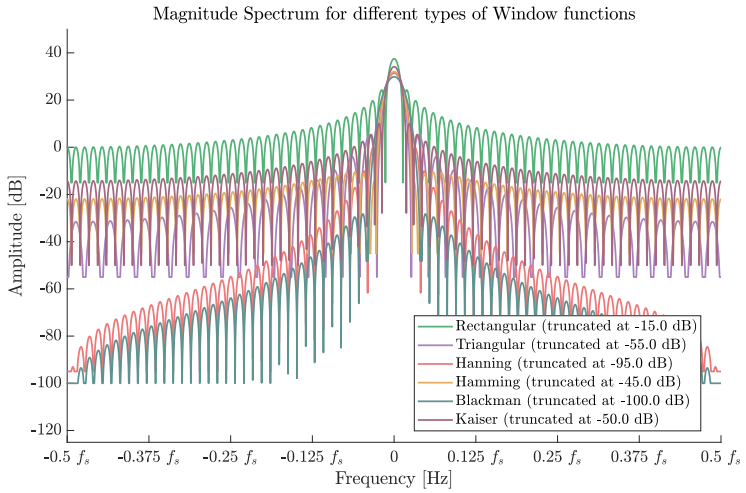



Figure 4.2: Plot showing magnitude spectrum for various window types. Truncation of the spectra is for visual purposes.

In this study, a Hanning window is used since it has the best narrow main lobe to low side lobe ratio. In other words, the best compromise between frequency resolution and spectral leakage. While this window is a good compromise, it is still not perfect. Therefore, a 50% overlap is chosen in order to prevent losing important frequency content in the spectrum. Overlapping, as the name suggests, means taking consecutive segments of the time data which are overlapping by the designated percentage of segment length. Figure A.4 gives a visual representation of window overlapping. Taking a higher overlap percentage is not wanted since that would require too many extra computations.

### 4.1.4 Feature Extraction

Now that the PSD is computed for each epoch, features can be extracted from these spectra. There are numerous different types of features that can characterise EEG signals [26] [36]. In this study the most prominent features of these have been selected. Most of them extract information from $\delta$ to $\beta$ the frequency bands, which were discussed in chapter 1. These features are:

- ***Mean Frequency (MNF)*** measures the average frequency value in the power spectrum and is computed as: $\frac{\sum_{k=1}^{N} f(k) \cdot P_{Welch}(k)}{\sum_{k=1}^{N} P_{Welch}(k)}$ where $P_{Welch}$ represents the power spectral density of an epoch and $f(k)$ represents the N frequency component appearing in the power spectrum..
- ***Band power*** represents the average power in each sub-band frequency and is computed as: $P_{abs} = \sum_{k=a}^{b} P_{Welch}(k)$
  where $a$ and $b$ represent the lower and upper limits of the frequency component of interest.

- ***Relative power per band:*** $P_\alpha, P_\theta, P_\beta$ and $P_\delta$ represent the ratio between the absolute power in each sub-band frequency and the total power.
- ***Power ratios*** computed as the ratio between each relative power band: $\frac{P_\delta}{P_\beta}, \frac{P_\delta}{P_\theta}, \frac{P_\delta}{P_\alpha}, \frac{P_\theta}{P_\beta}, \frac{P_\theta}{P_\alpha}, \frac{P_\alpha}{P_\beta}$
- ***Energy per band*** indicates the strength of the signal in the specific frequency band and is denoted as : $\sum_{k=a}^{b} \frac{|X(k)^2|}{N}$
  where $a$ and $b$ represent the lower and upper limit of the frequency components of interest.
- ***Energy ratios*** computed as the ratio between each energy band: $\frac{E_\delta}{E_\beta}, \frac{E_\delta}{E_\theta}, \frac{E_\delta}{E_\alpha}, \frac{E_\theta}{E_\beta}, \frac{E_\theta}{E_\alpha}, \frac{E_\alpha}{E_\beta}$
- ***Spectral Entropy (SE)*** estimates the uniformity of the signal energy distribution.[37] It is given by: $\sum_{k=1}^{N} p_{Welch}(k) \cdot \log_2(\frac{1}{p_{Welch}(k)})$ where $p_{Welch}(k)$ represents the normalised power at a certain frequency. From these SE's the standard deviations and the minimum values are computed.

This process results in 26 extracted features for each channel. Thus a total of 156 features are acquired when six channels are used.

## 4.2 Time-Frequency Domain Analysis

As the DFT forces a trade-off between time and frequency resolution, it is not ideal to analyse non-stationary signals like EEG's. To overcome this problem, the WT can be used to preserve time and frequency information. The WT is a method that divides data into different frequency components represented by coefficients [38].
WT makes use of a window that varies in frequency; at lower frequencies the window is broad, while at higher frequencies the window is much more narrow. There are two types of WT, the Continuous Wavelet Transform (CWT) and the DWT.

### 4.2.1 Continous Wavelet Transform

The CWT is computed as:

$$\text{CWT}(a,b) = \int_{-\infty}^{\infty} x(t) \cdot \psi^*_{a,b}(t)\, dt \tag{4.4}$$

where $x(t)$ represents an epoch and $\psi(t)$ is the varying analyzing window (called the mother wavelet). $a$ and $b$ denote the dilation factor and translation parameter applied to the mother wavelet respectively. This can be expressed as:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}) \tag{4.5}$$

This equations shows that the epochs are convolved with a set of wavelets containing different scaling factors, resulting in the CWT coefficients $C_{a,b}$.
The downside to this approach is that it must use every possible wavelet with different coefficients which is redundant and computationally impossible.

### 4.2.2 DWT

Similarly to the relation between the FT and the DFT, the DWT is a discretised version of the CWT. The scale and translation terms are set to a= $2^j$ and b=$k2^j$ where $k$ and $j$ are integers. Equation 4.5 then transforms into:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}}\psi(2^{-j}t - k) \tag{4.6}$$

With the DWT it is possible to get the wavelet coefficients of certain frequency bands from which features can be extracted.
The DWT can be implemented as a filter bank, which is a collection of filters that decomposes a signal into a set of frequency bands [39]. Figure 4.3 represents the four-level DWT decomposition applied on each epoch to obtain the coefficients for each frequency band.

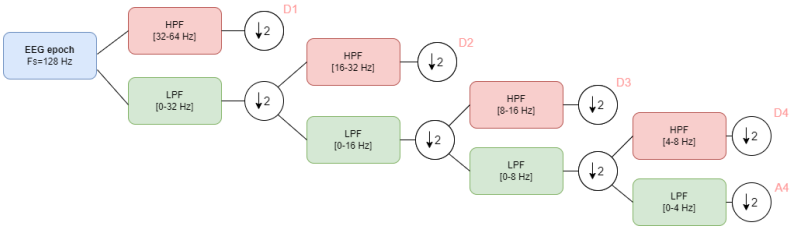



Figure 4.3: Flow chart for four level DWT.

The process starts at the first level where the EEG epoch signal goes into a band pass filter which is a combination of a High Pass Filter (HPF) and Low Pass Filter (LPF). Two corresponding coefficients are obtained, Approximation coefficient (A1) after the LPF and Detailed coefficient (D1) after the HPF. This process is repeated for A1 until the desired sub-band frequency coefficients are obtained. The decomposed signals D1-D4 and A4, shown in Figure 4.3 roughly correspond to the recognised brain signal frequency ranges $\theta$, $\alpha$, $\beta$ and $\gamma$ respectively. At each stage the output signal from the LPF is downsampled by two to decrease computation time. This is possible because half the frequencies are removed at the output of the LPF step. As explained in chapter 3, the signal can then be safely downsampled.

#### Selecting Mother Wavelet

Multiple mother wavelets can be used with the DWT such as the Shannon, Haar, Daubechies, and Morlet mother wavelets. Each of them gives different results for the DWT. It is therefore crucial to identify the wavelet that fits the data best. In this study the Daubcchies 4 (db4) mother wavelet is selected since it is widely used in biomedical signal processing and is proven to be the best choice for classifying epileptic seizures [40].

### 4.2.3 Feature Extraction

Feature extraction is then performed on the detailed and approximation coefficients. Again, the most prominent features presented in the literature have been used. The following statistical features are extracted from these wavelet coefficients: [41] [42] [43]

- ***Mean Absolute Value (MAV)*** gives the average absolute amplitude of the coefficients and is computed as: $\frac{1}{N}\sum_{j=1}^{N}|x_i|$ where N represents the number of samples contained in the wavelet coefficients $x_i$
- ***Average Power (AVP)*** measures the strength per unit time of the signals and is computed as: $\frac{1}{N}\sum_{j=1}^{N}|x_i^2|$
- ***Standard deviation ($\sigma$)*** measures the dispersion of the coefficients from its mean. It is given by: $\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu)^2}$ where $\mu$ represents the mean of the wavelet coefficients.
- ***Variance*** measures the statistical dispersion of a random variable and is given by: $\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu)^2$
- ***Mean*** indicates the average signal amplitude and is denoted as: $\frac{1}{N}\sum_{i=1}^{N}x_i$
- ***Energy*** indicates the strength of the signal and is denoted as: $\sum_{i=1}^{N}|x_i^2|$
- ***Kurtosis*** provides information on the shape of the distribution, more specifically it measures the degree of peakness in curves of the distribution. It is computed as: $\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^4}{\sigma^4}$

- ***Skewness*** also provides information on the shape of the distribution, more specifically it gives an indication on how much data is on one side of the mean or indicates the lack of symmetry. It is denoted as: $\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)}{\sigma^3}$

This process results in 8 features extracted from each of A4, D4, D3 and D2. Thus a total of 32 features for each channel results in a total of 216 extracted features for the six channels.

### 4.2.4 Normalization

It is crucial to perform feature scaling on the feature matrices prior to performing Principal Component Analysis (PCA). If there are large differences between the ranges of features, it becomes much harder to properly compare them when using PCA. Normalization can be used to make features much more comparable, it shifts and scales all features such that they end up ranging between 0 and 1. Equation 4.7 gives the formula used for normalization:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.7}$$

Where $X$ represents a vector of features.

## 4.3 Dimensionality Reduction

PCA is a method used to reduce the number of features while still retaining most of the information from the original data set. It transforms a high dimensional feature matrix to a lower dimensional orthogonal feature space. PCA does this by constructing linear combinations of the original features, called principal components [44]. Reducing dimensions means that training the model is significantly less computationally complex. The newly created features or principal components are characterised by their explained variance. The higher the explained variance of that principal component, the more information it contains from the original feature matrix.
PCA is applied on the one and six channel feature matrices obtained through frequency and time-frequency domain analysis of the EEG data. Figure 4.4 shows the two principal component sets with their explained variance, obtained from a single channel of EEG epochs.

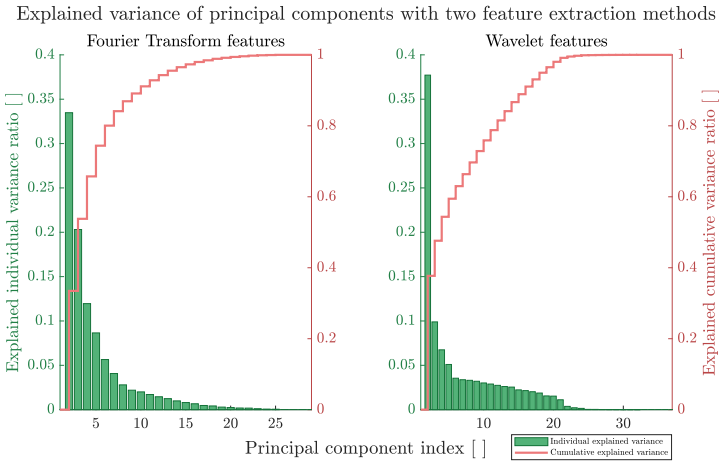



Figure 4.4: Single channel explained variance for each principal component.

As one can see in Figure 4.4, the individual explained variance score becomes extremely low for the last indices. These can be neglected for a reduction in dimensionality, while not losing important information. A simple procedure to decide which principal components to neglect can be done by setting a threshold

for the cumulative explained variance. The first $n$ principal components are taken until this threshold is reached. A threshold of 95% and 99% have been selected for further analysis.
This process results in two diminished feature vectors for each case. Table A.2 and Table A.3 depict the feature vector spaces for one and six channels after applying PCA with a threshold of 95% and 99%.

# Chapter 5

# Classification

Seizures can now be classified using the features extracted from the epochs EEG's in chapter 4. This can be done by manually checking the features for certain thresholds that need to be exceeded to classify it as a seizure. This can only be done when there are clear distinguishing features. Seizures create abnormal patterns of varying length, making it unreasonable to use this method. Instead, supervised machine learning will be used to classify seizures. This chapter will discuss Machine Learning and ML models that can be used to classify seizures.

## 5.1 Machine Learning

Machine learning is a part of Artificial Intelligence (AI) that focuses on building a model based on (large amounts of) data in order to make predictions or decisions without explicitly being programmed to do so [45]. There are multiple approaches that can be used, such as supervised and unsupervised learning. In supervised learning the machine is tasked with finding patterns that connect the input data to their classification labels; the machine finds the features that are best at predicting these labels [46]. On the other hand, in unsupervised learning the input is unlabelled and the machine is tasked with finding patterns in the input data in order to create classifications for this input [47]. As the timestamps of the seizures are already known, supervised models will be used in this study.
Based on the above, one would expect that inputting as much data as possible into a ML model would improve the classification results. This is unfortunately only partly true as there are some pitfalls, such as overfitting. This is the scenario in which the model trains too much on the training data. The result is a model that only works properly on the training data, while performing much worse on unseen data. It could also be the case that the features are not representative, making the model perform worse.

### 5.1.1 Hyperparameter Tuning

In ML a hyperparameter is a variable that is used to control the learning process [48]. These hyperparameters are selected before training and stay constant during training.
Hyperparameter tuning is the process of selecting the most optimal hyperparameters. There are different approaches for tuning; in this study Bayesian optimisation is used. This method tries to find the ideal hyperparameters by using prior and posterior probabilities [49]. The 'best' hyperparameters are those that give the lowest error rate. By training the model each time with different hyperparameters it seeks the ones that give the least amount of errors. For training 50 iterations of tuning are performed, which should be more than enough to reach close to optimum hyperparameters. As hyperparameter tuning quickly converges making only very small gains after the first 20 iterations of tuning. The specific hyperparameters that are tuned will be specified per ML model.

## 5.2 Data Preparation

A few measures were taken to make sure the machine learning models did not overfit. Only files with recordings containing seizures were used from the CHB-MIT data set. Only a small part of the hour long recordings consist of labelled seizures; the average length of seizure data is around 40 seconds per recording. This means only 1% of the data contains a seizure; including data sets that have zero seizures will only skew the training towards classifying everything as a non-seizure.
Ten-fold cross validation is done to ensure that the model does not overfit on the training data [50]. This splits the data into a training and a testing set. 90% of the data is used for training the model, with the remaining 10% used for testing the model. This is repeated 10 times and each iteration a different 10% of the data is selected. The final split is an average of these ten iterations.

## 5.3 Machine Learning Models

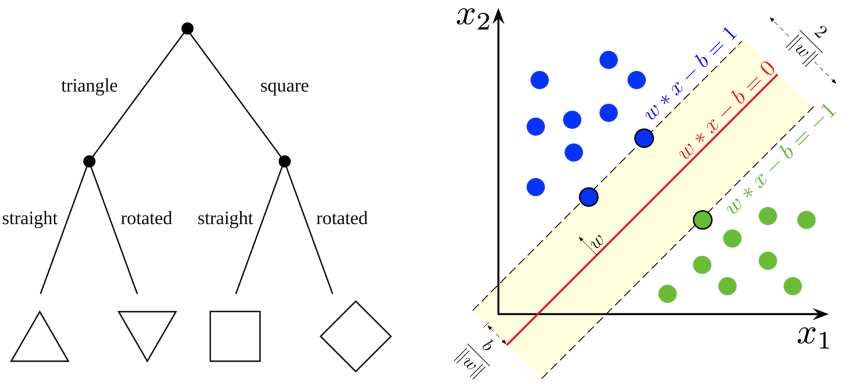
Different ML models have different strengths and multiple models were tested to see which one performed best for classifying epileptic seizures. In this study, the models tested are the random forest model, the Support Vector Machine (SVM) and the classifier Neural Network (NN).

### 5.3.1 Random Forest

A decision tree is a type of model that uses a tree-like structure to visualise reasons for decisions and their consequences. They are very popular in machine learning since they are simple and relatively easy to understand [51]. In ML there are two main types of decision trees: classification and regression trees. The former is useful when predicting classes, while the latter is used when predicting continuous variables. Classification trees are obviously more useful when classifying seizures.
In Figure 5.1a an example of a decision tree is shown. A classifier can use input features to go through the tree node per node until a classification result or another stop condition is reached. These trees are great for simplifying a complex classification problem, but they are not perfect. They are very prone to overfitting when too many nodes are used as the rules per branch could become too specific.
Random forests can be used to counteract the overfitting. These forests contain multiple decision trees that each use a random subset of the features from the training data. Therefore, it will not classify based on a single feature, but on a larger set. To make predictions based on test data, it uses unweighted voting from the results of each tree to decide the final result.
This method is mainly used as it does not require any hyperparameter tuning. This is because the hyperparameters define how the model should learn from the features, as the features change for every tree no overarching hyperparameters can be set to optimize every tree. Ten-fold cross validation is not performed here as the model already takes its own subsets of the data. This means that it performs cross validation already when constructing the final model.

### 5.3.2 Support Vector Machine

An SVM attempts to separate classes in a feature space using a hyperplane [54]. This hyperplane should separate as much of the training data as possible.
Figure 5.1b shows an example of a hyperplane that separates features in a two dimensional feature space. The name SVM comes from the support vectors that are used to find the optimal hyperplane. In Figure 5.1b they are visualised as dashed lines at $w \cdot x - b = \pm 1$. The optimal hyperplane maximises the minimum distance between data points and the hyperplane, while at the same time containing the smallest amount of errors. Since this optimum is not always possible, a cost factor is introduced that controls the amount of data points that are allowed on the 'wrong side' of the hyperplane. In Figure 5.1b this would be visualised as a few blue dots in between the green dots, while keeping the same hyperplane. A large cost factor will make the model try to classify all training points correctly, which is very prone to overfitting. A small cost factor however would result in a wrong classification for a large part of the data. There is a sweet spot for




(a) Decision tree [52]. (b) SVM hyperplane in two dimensions [53].

Figure 5.1: Visual representation of ML models

the cost factor, which can be found by doing hyperparameter tuning.

A problem with SVM is that the hyperplane is linear. Data that is not linearly separable is therefore hard to classify using an SVM, but this can be solved by Kernels. Kernels perform a non-linear transformation on the input features, which allows the hyperplane to be drawn through these new dimensions.

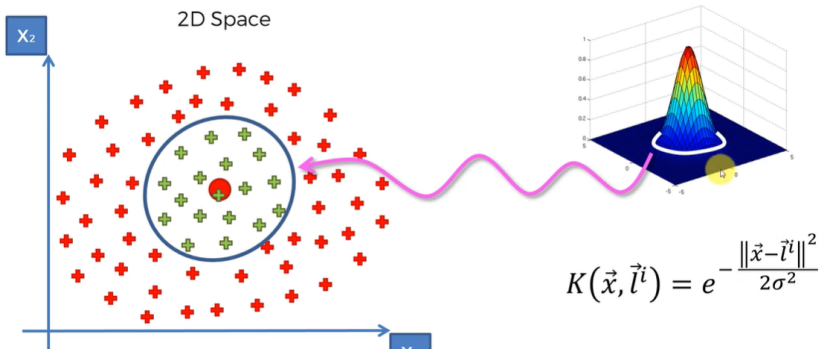



Figure 5.2: Example of Radial Basis Function (RBF) kernel applied to data that is not linearly separable.

In this research, an RBF kernel – shown in Figure 5.2 – was used to construct the classifier. An RBF is a function whose value relies on some fixed point and the distance to that fixed point [55]. This was chosen as visual inspection showed no clear linear separation between the features.
Weights should be adjusted such that misclassifying seizures is kept to a minimum, while wrongly classifying non-seizures is penalised less. This method has also been previously used in EEG epilepsy classification with good results [56].

### 5.3.3 Neural Networks

The NN is a concept inspired by the way human brains operate [57]. It makes use of artificial neurons, sometimes called perceptrons. In the context of ML a neuron holds a single value between 0 and 1. A NN consists of multiple neurons that are separated in layers, usually these are the input, hidden and output layers. The classification result is then decided by the neuron in the output layer that contains the largest value. The neurons in the input layer consists of all the input features. The values the neurons hold in the other layers are determined by a function that uses the weighted sum of all neurons in the previous layer as an argument. This function is called an activation function. A visual representation of such a neural network is shown below:

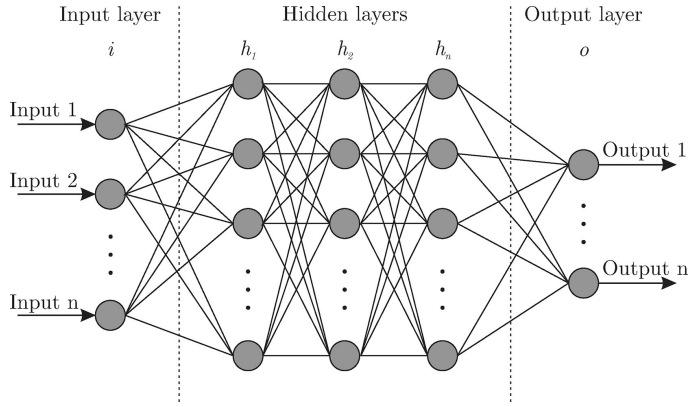



Figure 5.3: Visual representation of a fully connected neural network[58].

In a neural network, the input layer and output layer have a fixed size. As mentioned previously, the amount of neurons in the input layer are determined by the number of input features. The output layer's size is set by the amount of classification labels. In our case there are only two classification results – a seizure or no seizure – so the output layer contains two neurons. In contrast, the hidden layers can be any size. Larger sizes need more computation time but generally give more accurate classification results.
As mentioned previously, the values of the neurons not in the input layer are determined by their activation function that uses weighted sum of the neurons in the previous layer. During training, these weights are adjusted such that the final classification result is as accurate as possible. To make sure that the model performs better for this specific scenario, hyperparameter tuning is performed. During the tuning, the following parameters are determined: learning rate, hidden layer size and activation functions for the neurons. The NN has also shown promise in previous studies for scalp-EEG classification, giving further reason to try this method [59, 60].

# Chapter 6

# Discussion of Results

## 6.1 Results

In chapter 3, chapter 4 and chapter 5 methods to do pre-processing, feature extraction and classification of seizures respectively were discussed. These methods were implemented and tested and this chapter will discuss the obtained results of classification.

### 6.1.1 Analysed data

Only a subset of patients in the CHB-MIT data set was used for the analysis. The largest group excluded are the patients numbered 11 and above. This is done because their measurement setups are different – sometimes even changing between recordings – and therefore the resulting measurements are hard to compare with the other patients. Furthermore, the second patient was also excluded as their recordings only contain two seizures. This patient had 30 epochs containing seizure data, which is too little to train on; ten-fold cross validation would only train on 3 epochs per iteration. In the end only patient 1 and patients 3 up to 10 are left and used to get the results. They all use the same layout for the electrodes and contain at least 100 epochs of seizure data. Patient 6 is the only outlier for this as their recordings contain only 60 epochs of seizure, however, those were spread over 10 unique seizures. This gave an average length of 18 seconds per seizure, a unique outlier in the data set.

### 6.1.2 Epoch Length

The epoch length needs to be selected before running any of the models with hyperparameter tuning. As already specified in chapter 3, there is an upper and lower bound for the length, which are 3 and 6 seconds respectively. Adding this variable to the hyperparameter tuning would exponentially drive up computation time. It was opted for to test it separately on a randomly selected patient, in this case patient 3. The random forest was used with standard hyperparameters and the FT related features were used. This model performs well without hyperparameter tuning, making it a good model to test epoch length on.

| EpochLength (s) | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| Accuracy (%) | 99.7 | 99.6 | 99.6 | 99.6 |
| Sensitivity (%) | 86.9 | 79.7 | 80.3 | 84.5 |

Table 6.1: The accuracy and sensitivity for different epoch lengths using the random forest with 100 trees on the data of patient 3.

Table 6.1 shows that with default MATLAB hyperparameters a length of 3 seconds is optimal. This was taken as the epoch length for all further analysis.

### 6.1.3 Hyperparameters

With the epoch length known, hyperparameter tuning was done for the SVM and NN models, both per patient and for all patient data combined. The hyperparameters for per patient data was tuned on a randomly selected patient – the 5$^{th}$ one – in order to generalise them. The resulting hyperparameters for SVM and NN can be found in section B.1.

## 6.2 Results Per Patient

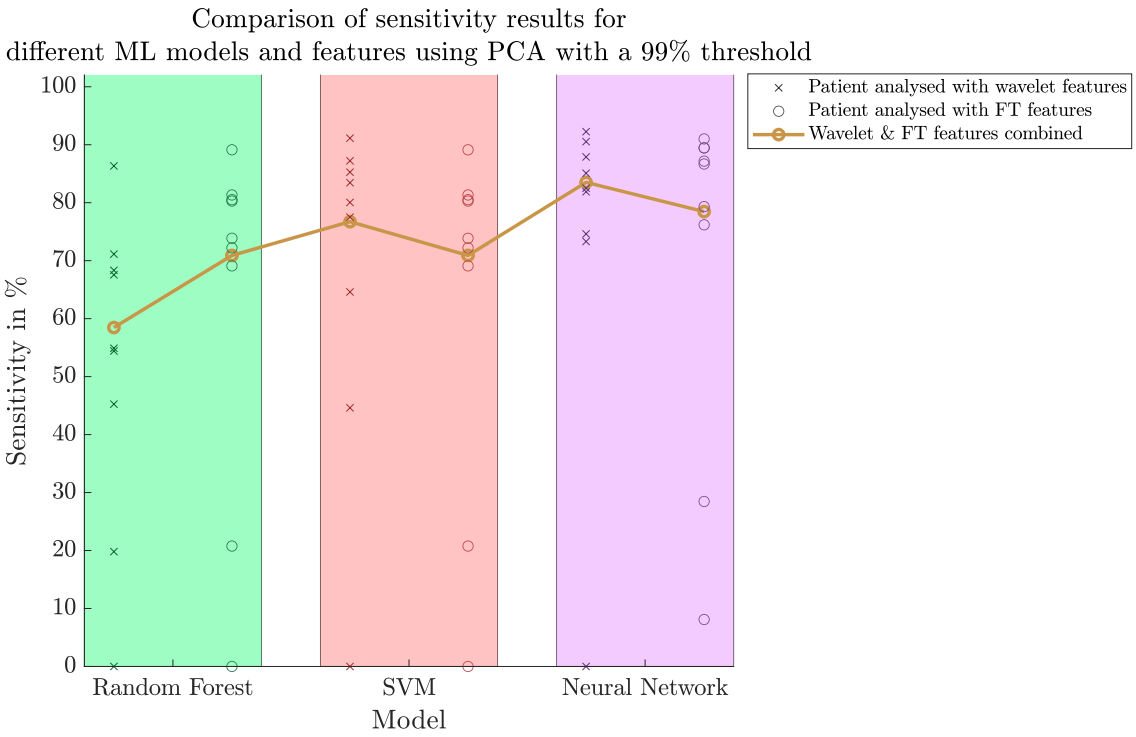



Figure 6.1: Results of classification sensitivity for three ML models. Each data point in the figure is the sensitivity result of a single patient.

In Figure 6.1 the sensitivity results of all previously discussed ML models are shown. The features were extracted from both the WT and FT and are indicated with different markers. In section B.2 the full results are visible, along with an additional graph showing the results for a threshold of 95% for PCA and no PCA at all. Furthermore, patient 6 is excluded from the mean, as their classification is a consistent outlier with a 0% sensitivity, and is elaborated more on later.
From the results it can be seen that it is possible to detect seizures using scalp-EEG with only channels close to the ear. While the sensitivity varies between features and ML models, it is still significant enough to classify most seizures correctly. The best performance comes from a NN using features extracted from the WT; a sensitivity of 83.5% is reached. Also, the analysis performed using the WT consistently performs better than the FT, as expected [34].
The reason this is not the case for the random forest in Figure 6.1 is due to the PCA. A random forest selects a subset of the input features, meaning that with PCA in this ML model a subset of principal components is taken. This means that a random forest could take principal components with a low explained variance, creating a bad classifier. Figure A.6 shows that without PCA, the trend of WT features outperforming FT features holds also for the random forest.

## 6.3 Results of Grouped Patients

The training and classifying based on a data set consisting of multiple patients showed a worse performance than the data sets per patient. While the NN did reach a sensitivity of 81.9% and an accuracy of 99.7%, it was still lower than that of the results per patient with a NN. The other methods performed even worse in this task, with only the SVM getting a sensitivity of 56.9% and an accuracy of 99.4%. The random forest getting a sensitivity of 53.8% and accuracy of 99.4%.

## 6.4 Processing time

One of the requirements is that time it takes to classify an epoch is shorter than the actual epoch itself. Timings were performed inside MATLAB to measure how long it takes to do feature extraction and classifying. This was measured on an i5-6500 Central Processing Unit (CPU), with 8 GB of Random Access Memory (RAM). The pre-processing with ASR took $1.23$ ms for a single epoch, and $0.27$ ms without ASR. The feature extraction for a single epoch takes $1.7$ ms on average, for six channels it takes $10.2$ ms. The classification with an already trained model takes $0.26$ ms for a single epoch, if no PCA is applied. With PCA it is an extremely tiny bit faster. This summed together gives that a single epoch takes roughly $13.5$ ms to be processed and classified. While this is not directly representative of the timings when it is implemented in hardware, it does show that the processing time is orders of magnitude smaller than the epoch length.

## 6.5 Discussion

From the results per patient it becomes clear that the sixth patient is a clear outlier. Removing them from the average sensitivity for each test improves the sensitivity average by 6 to 7%. There are multiple reasons as to why this patient performs worse. From a visual comparison of Figure A.12 and Figure A.13 it becomes apparent that the patient's EEG has a lot of irregular patterns. The channels near the ear especially suffer from this; there is no clear visual distinguishing factor between seizure and non seizure epochs. This could potentially be attributed to the fact that the patient is the youngest of the data set, at an age of 1.5 years old during recordings. The extracted features from the EEG data show no distinguishing results; potentially there are other features that are more suited for this patient's EEG. For example, the dominant spectral component which was not extracted in this research, but which was extracted in the original study associated with this data set [17]. This feature describes which frequencies have the largest peaks in the frequency domain.
The results do not reach the level of previous work for scalp-EEG. For example, one paper that uses the same data set achieves a 96% sensitivity with per patient analysis. They tested 163 seizures [17] and even got good results for the sixth patient. This could be due to the access to all other channels, instead of just those close to the ear. However, for ear-EEG the performance is somewhat comparable with other studies. Another research using behind the ear-EEG reaches an average sensitivity of 82.2% [6], which we also reach when the sixth patient data is excluded. The bad performance of the SVM was expected, as the features were not well separable in a feature space. This can be seen in Figure A.5 and the same trend also held for other combination of features. It was almost impossible to separate the classes in a feature space well by drawing a straight line. The reason could be the limited number of channels; the selected features might not be as distinctive in them. This is probably not the entire story, as the other ML models did perform better using the exact same features. Choosing a different kernel function might help improve accuracy and sensitivity results, as maybe the RBF kernel was not the best choice in this scenario.
Some caution must be reserved regarding the results, considering the used data is not actual ear-EEG but a set of scalp-EEG channels. We hypothesize that the signal to noise ratio is lower with the ear electrodes, due to ear-EEG electrodes being smaller in general. This could potentially lower the accuracy and sensitivity of classification. The location of the electrodes changing could also influence what artefacts are measured, further altering the results.

# Chapter 7

# Conclusion

It is already possible to detect epileptic seizures using scalp-EEG with a reasonable accuracy. For ear-EEG this is has not yet been researched as thoroughly. In order to make progress in this field, we have attempted to classify epileptic seizures using a simulation of ear-EEG. This simulation was created by selecting scalp-EEG electrodes close to the ear. While this method of attempting to approximate ear-EEG is not perfect, the results obtained do showcase the possibility of detecting seizures with a reduced amount of channels. In this study we have found that with a NN model, a sensitivity of 83.5% can be reached. While this result is lower than other comparable scalp-EEG based studies – most likely due to the reduced number of channels – it is close to other ear-EEG results.
The detection has been done using purely EEG with channels placed close to the ears. The chosen length of epochs and thus length of recordings is three seconds. On average, the reached accuracy is larger than 99%. In contrast, the average sensitivity has not surpassed 95% yet. Looking back at the programme of requirements, this means that this epilepsy detection system is not adequate enough to be used in combination with t-VNS in a closed loop system. Furthermore, there are some requirements that have not been studied due to time constraints, such as the hardware implementation and the control of the stimulus circuit. More work is required to improve the sensitivity of seizure detection and to create a hardware implementation before ear-EEG can be used to detect epilepsy on a day-to-day basis. Some potential improvements for the future will discussed in the next section

## 7.1 Future Work

With the restricted time of 9 weeks to conduct our research, choices had to be made regarding the techniques to use for filtering, feature extraction and classification. Not everything could be attempted and, along the project, new methods were discovered which could have resulted in better outcomes.
Most importantly, this study suggests that detecting epileptic seizures using in-ear EEG could be achievable. However, it is crucial to apply the algorithm on actual data recorded from electrodes inside the ear in order to properly quantify its performance.
We have only focused on detecting the ictal phase of an epileptic seizure. Predicting if a seizure is going to take place requires the ability to also classify the preictal phase. In the future additional research on what features of EEG's are good at classifying this phase should be done.
Most of the used features were extracted from the four lowest sub-band frequencies detailed in Table A.1. However, epileptic seizures could arise differently within these sub bands, which could mean that smaller sub-band frequencies should be analysed as well [6].
PCA was used as a technique to reduce the feature vector size, but other techniques should be used – with or without PCA – such as Independent Component Analysis (ICA) [61]. Moreover, a feature selection techniques such as Minimum Redundancy Maximum Relevance (mRMR) should also be looked at.
Finally, different ML methods should also be researched. A vast range of methods have been tried to classify seizures in scalp-EEG, which is not limited to those discussed in this study. The used models

should also be adjusted if necessary, as other kernels could be used for the SVM and other structures for the NN.
Storage of the recorded data is one subject this study has not looked into, but is very important to create a portable ear-EEG device. When recording an EEG with a set epoch time of three seconds there is a possibility of only partially recording a seizure, meaning it is likely to be misclassified. It could be useful store several epochs at a given time with techniques that give more weight to the most recent epoch such as in [62].
The use of ASR is also very important to consider. While it does reduce EMG artefacts, something that is very important to do for a device that is meant to classify seizures all day long, it is also very expensive computationally. It would be best to look at alternative methods that reduce EMG artefacts for less computational cost, or improve the ASR's efficiency. Furthermore, if ASR is going to be used it is important to think about the memory storage needed for a reference signal that is required to perform the technique.
Despite the research in this study being incomplete, it is a good stepping stone for future research in the field of epileptic seizure detection using ear-EEG. There is still a lot of untapped potential in this new technology and tapping into that could improve the lives of many epileptic people around the world.

# List of abbreviations

**AC** Alternating Current

**AI** Artificial Intelligence

**ASR** Artefact Subspace Reduction

**AVP** Average Power

**bpm** beats per minute

**CDS** Correlated Double Sampling

**CPU** Central Processing Unit

**CWT** Continuous Wavelet Transform

**db4** Daubcchies 4

**DC** Direct Current

**DFT** Discrete Fourier Transform

**DWT** Discrete Wavelet Transform

**ECG** Electrocardiography

**EEG** Electroencephalography

**EMG** Electromyography

**FFT** Fast Fourier Transform

**FIR** Finite Impulse Response

**FT** Fourier Transform

**HPF** High Pass Filter

**ICA** Independent Component Analysis

**LPF** Low Pass Filter

**MAV** Mean Absolute Value

**ML** Machine Learning

**MNF** Mean Frequency

**mRMR** Minimum Redundancy Maximum Relevance

**NN** Neural Network

**PCA** Principal Component Analysis

**PCB** Printed Circuit Board

**PSD** Power Spectral Density

**RAM** Random Access Memory

**RBF** Radial Basis Function

**SE** Spectral Entropy

**SVM** Support Vector Machine

**t-VNS** Transcutaneous Vagus Nerve Stimulation

**VNS** Vagus Nerve Stimulation

**WT** Wavelet Transform

# Bibliography


[1] S. L. Moshé, E. Perucca, P. Ryvlin, and T. Tomson, "Epilepsy: new advances," *The Lancet*, vol. 385, pp. 884–898, Mar. 2015.

[2] NHS, "Treatment of epilepsy," 9 2020. `https://www.nhs.uk/conditions/epilepsy/treatment/`, accessed 6th June 2022.

[3] W. Löscher, H. Potschka, S. M. Sisodiya, and A. Vezzani, "Drug resistance in epilepsy: Clinical impact, potential mechanisms, and new innovative treatment options," *Pharmacological Reviews*, vol. 72, pp. 606–638, June 2020.

[4] J. Engel, "The current place of epilepsy surgery," *Current Opinion in Neurology*, vol. 31, pp. 192–197, Apr. 2018.

[5] J. Y. Y. Yap, C. Keatch, E. Lambert, W. Woods, P. R. Stoddart, and T. Kameneva, "Critical review of transcutaneous vagus nerve stimulation: Challenges for translation to clinical practice," *Frontiers in Neuroscience*, vol. 14, Apr. 2020.

[6] Y. Gu, E. Cleeren, J. Dan, K. Claes, W. V. Paesschen, S. V. Huffel, and B. Hunyadi, "Comparison between scalp EEG and behind-the-ear EEG for development of a wearable seizure detection system for patients with focal epilepsy," *Sensors*, vol. 18, p. 29, Dec. 2017.

[7] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, and O. A. Alomari, "Person identification using EEG channel selection with hybrid flower pollination algorithm," *Pattern Recognition*, vol. 105, p. 107393, Sept. 2020.

[8] J. A. Urigüen and B. Garcia-Zapirain, "Eeg artifact removal—state-of-the-art and guidelines," *Journal of neural engineering*, vol. 12, no. 3, p. 031001, 2015.

[9] A. Medvedev, V. Temkina, A. Makaryan, E. Sivolenko, B. Hovhannisyan, and H. Ayvazyan, "Extracting human brain signals from the EEG records using LabVIEW and advanced signal processing," in *Springer Proceedings in Physics*, pp. 183–190, Springer International Publishing, 2022.

[10] T. Nakamura, Y. D. Alqurashi, M. J. Morrell, and D. P. Mandic, "Automatic detection of drowsiness using in-ear EEG," in *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, July 2018.

[11] T. Nakamura, Y. D. Alqurashi, M. J. Morrell, and D. P. Mandic, "Hearables: Automatic overnight sleep monitoring with standardized in-ear EEG sensor," *IEEE Transactions on Biomedical Engineering*, vol. 67, pp. 203–212, Jan. 2020.

[12] D.-H. Jeong and J. Jeong, "In-ear EEG based attention state classification using echo state network," *Brain Sciences*, vol. 10, p. 321, May 2020.

[13] Swartz Center for Computational Neuroscience, "EEGLAB." `https://sccn.ucsd.edu/eeglab/index.php`, accessed 9th June 2022.

[14] Mayo Clinic, "Symptoms and causes of seizures," Feb 2021. https://www.mayoclinic.org/diseases-conditions/seizure/symptoms-causes/syc-20365711, accessed 9th June 2022.

[15] A. B. Usakli, "Improvement of eeg signal acquisition: An electrical aspect for state of the art of front end," *Computational intelligence and neuroscience*, vol. 2010, 2010.

[16] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[17] A. H. Shoeb, *Application of machine learning to epileptic seizure onset detection and treatment*. PhD thesis, Massachusetts Institute of Technology, 2009.

[18] H. H. Jasper, "The ten-twenty electrode system of the international federation," *Electroencephalogr. Clin. Neurophysiol.*, vol. 10, pp. 370–375, 1958.

[19] R. W. Homan, J. Herman, and P. Purdy, "Cerebral location of international 10–20 system electrode placement," *Electroencephalography and clinical neurophysiology*, vol. 66, no. 4, pp. 376–382, 1987.

[20] S. L. Kappel, D. Looney, D. P. Mandic, and P. Kidmose, "Physiological artifacts in scalp EEG and ear-EEG," *BioMedical Engineering OnLine*, vol. 16, Aug. 2017.

[21] V. Goverdovsky, W. Von Rosenberg, T. Nakamura, D. Looney, D. J. Sharp, C. Papavassiliou, M. J. Morrell, and D. P. Mandic, "Hearables: Multimodal physiological in-ear sensing," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[22] M. G. Bleichner, *Studying individual noise disturbance using long term ear-EEG (electroencephalography) recordings in everyday life*. Universitätsbibliothek der RWTH Aachen, 2019.

[23] Harvard Health, "What your heart rate is telling you," Aug 2020. https://www.health.harvard.edu/heart-health/what-your-heart-rate-is-telling-you, accessed 9th June 2022.

[24] I. Zibrandtsen, P. Kidmose, and T. Kjaer, "Detection of generalized tonic-clonic seizures from ear-eeg based on emg analysis," *Seizure*, vol. 59, pp. 54–59, 2018.

[25] C. C. Enz and G. C. Temes, "Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling, and chopper stabilization," *Proceedings of the IEEE*, vol. 84, no. 11, pp. 1584–1614, 1996.

[26] A. Khosla, P. Khandnor, and T. Chand, "A comparative analysis of signal processing and classification methods for different applications based on eeg signals," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 2, pp. 649–690, 2020.

[27] T. R. Mullen, C. A. Kothe, Y. M. Chi, A. Ojeda, T. Kerth, S. Makeig, T.-P. Jung, and G. Cauwenberghs, "Real-time neuroimaging and cognitive monitoring using wearable dry eeg," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2553–2567, 2015.

[28] S. Blum, N. S. Jacobsen, M. G. Bleichner, and S. Debener, "A riemannian modification of artifact subspace reconstruction for eeg artifact handling," *Frontiers in human neuroscience*, p. 141, 2019.

[29] A. V. Oppenheim, J. R. Buck, and R. W. Schafer, *Discrete-time signal processing. Vol. 2*. Upper Saddle River, NJ: Prentice Hall, 2001.

[30] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[31] D.-W. Kim and C.-H. Im, "EEG spectral analysis," in *Biological and Medical Physics, Biomedical Engineering*, pp. 35–53, Springer Singapore, 2018.

[32] A. S. Al-Fahoum and A. A. Al-Fraihat, "Methods of EEG signal features extraction using linear analysis in frequency and time-frequency domains," *ISRN Neuroscience*, vol. 2014, pp. 1–7, Feb. 2014.

[33] O. Dressler, G. Schneider, G. Stockmanns, and E. Kochs, "Awareness and the EEG power spectrum: analysis of frequencies," *British Journal of Anaesthesia*, vol. 93, pp. 806–809, Dec. 2004.

[34] M. Kıymık, İ. Güler, A. Dizibüyük, and M. Akın, "Comparison of STFT and wavelet transform methods in determining epileptic seizure activity in EEG signals for real-time application," *Computers in Biology and Medicine*, vol. 35, pp. 603–616, Oct. 2005.

[35] D.-J. Jwo, W.-Y. Chang, and I.-H. Wu, "Windowing techniques, the welch method for improvement of power spectrum estimation," *Computers, Materials &amp Continua*, vol. 67, no. 3, pp. 3983–4003, 2021.

[36] N. Michielli, U. R. Acharya, and F. Molinari, "Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals," *Computers in Biology and Medicine*, vol. 106, pp. 71–81, Mar. 2019.

[37] "Detection, classification, and estimation in the ( t , f ) domain," in *Time-Frequency Signal Analysis and Processing*, pp. 693–743, Elsevier, 2016.

[38] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Jan. 1992.

[39] A. H. Shoeb and G. D. Clifford, "Chapter 16-wavelets ; multiscale activity in physiological signals c © 2006,"

[40] M. S. Fathillah, R. Jaafar, K. Chellappan, and R. Remli, "A study on eeg signals during eye-closed and eye-open using discrete wavelet transform," in *2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 674–678, 2016.

[41] D. V. G. S. and, "Electroencephalogram (EEG), its processing and feature extraction," *International Journal of Engineering Research and*, vol. V9, June 2020.

[42] M. Sameer and B. Gupta, "Detection of epileptical seizures based on alpha band statistical features," *Wireless Personal Communications*, vol. 115, pp. 909–925, June 2020.

[43] C. Sudalaimani, N. Sivakumaran, T. T. Elizabeth, and V. S. Rominus, "Automated detection of the preseizure state in EEG signal using neural networks," *Biocybernetics and Biomedical Engineering*, vol. 39, pp. 160–175, Jan. 2019.

[44] I. Jolliffe, "Principal component analysis," in *International Encyclopedia of Statistical Science*, pp. 1094–1096, Springer Berlin Heidelberg, 2011.

[45] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, "Automated design of both the topology and sizing of analog electrical circuits using genetic programming," in *Artificial Intelligence in Design'96*, pp. 151–170, Springer, 1996.

[46] J. Stuart *et al.*, "Artificial intelligence a modern approach third edition," 2010.

[47] G. Hinton and T. J. Sejnowski, *Unsupervised learning: foundations of neural computation*. MIT press, 1999.

[48] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

[49] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[50] R. O. Duda, P. E. Hart, *et al.*, *Pattern classification*. John Wiley & Sons, 2006.

[51] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, *et al.*, "Top 10 algorithms in data mining," *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.

[52] CFI, "Decision Tree." `https://corporatefinanceinstitute.com/resources/knowledge/other/decision-tree/`, accessed 9th June 2022.

[53] Wikipedia, "Support-vector machine — Wikipedia, the free encyclopedia." `http://en.wikipedia.org/w/index.php?title=Support-vector%20machine&oldid=1091818704`, 2022. [Online; accessed 10-June-2022].

[54] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[55] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," tech. rep., Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[56] U. R. Acharya, S. V. Sree, G. Swapna, R. J. Martis, and J. S. Suri, "Automated eeg analysis of epilepsy: a review," *Knowledge-Based Systems*, vol. 45, pp. 147–165, 2013.

[57] C. M. Bishop, "Neural networks and their applications," *Review of scientific instruments*, vol. 65, no. 6, pp. 1803–1832, 1994.

[58] F. Bre, J. M. Gimenez, and V. D. Fachinotti, "Prediction of wind pressure coefficients on building surfaces using artificial neural networks," *Energy and Buildings*, vol. 158, pp. 1429–1441, 2018.

[59] Y. Kumar, M. Dewal, and R. Anand, "Epileptic seizures detection in eeg using dwt-based apen and artificial neural network," *Signal, Image and Video Processing*, vol. 8, no. 7, pp. 1323–1334, 2014.

[60] A. Alkan, E. Koklukaya, and A. Subasi, "Automatic seizure detection in eeg using logistic regression and artificial neural network," *Journal of neuroscience methods*, vol. 148, no. 2, pp. 167–176, 2005.

[61] A. Subasi and M. I. Gursoy, "EEG signal classification using PCA, ICA, LDA and support vector machines," *Expert Systems with Applications*, vol. 37, pp. 8659–8666, Dec. 2010.

[62] G. Cormode, F. Korn, and S. Tirthapura, "Exponentially decayed aggregates on data streams," in *2008 IEEE 24th International Conference on Data Engineering*, pp. 1379–1381, IEEE, 2008.

# Appendix A

# Additional Figures & Tables

## A.1 Background Information

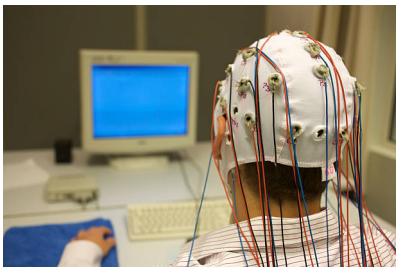
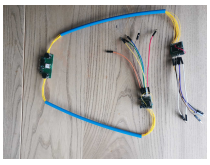


Figure A.1: Scalp-EEG measurements being conducted

| EEG waves | Related brain activity |
|---|---|
| $\delta$ (0.5-4Hz) | Deep (dreamless) sleep. |
| $\theta$ (4-8Hz) | Relaxation. |
| $\alpha$ (8-13Hz) | Occipital activity. |
| $\beta$ (13-32Hz) | Wakefulness and concentration. |
| $\gamma$ (25-140Hz) | Cognitive functioning, learning memory and information processing |

Table A.1: The frequency bands associated with EEG signals



(a) Full prototype

(b) Printed Circuit Board (PCB) used in the prototype

Figure A.2: Prototype of the ear-EEG device developed by P. Burgar and C. Weustink

## A.2 Pre-Processing

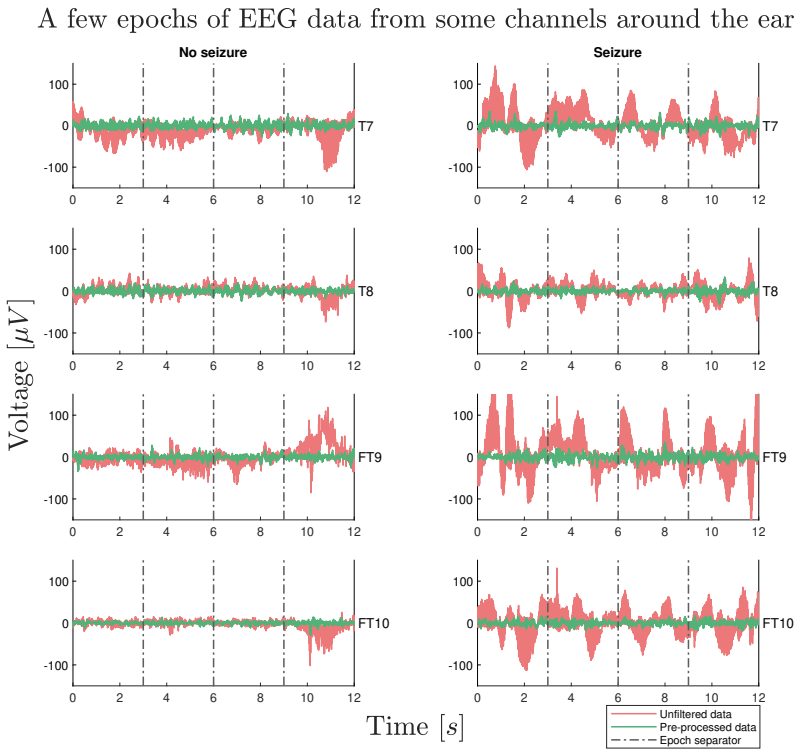



Figure A.3: Example of epochs of all used channels from a single patient. The figures on the left show epochs without seizures and the figures on the right show epochs containing seizures.

## A.3 Feature Extraction

### A.3.1 Overlapping windows

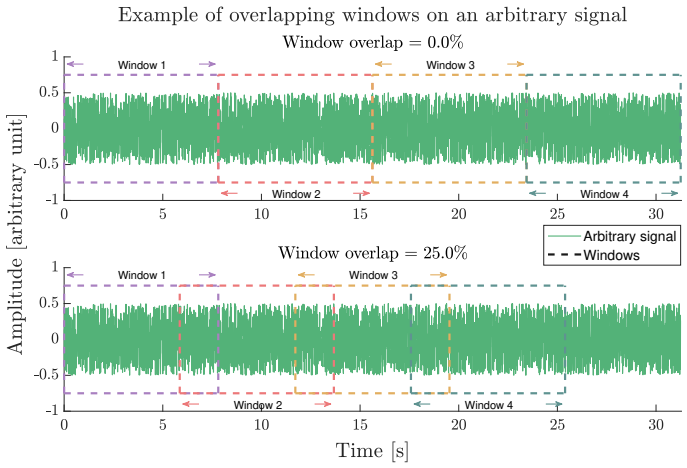



Figure A.4: Figure showing overlapping windows. The signal used does not represent an EEG.

### A.3.2 Feature dimensions

| 1 Channel | Without PCA | PCA (95%) | PCA (99%) |
|---|---|---|---|
| Feature Vector Fourier Dimension | 26x1 | 7x1 | 11x1 |
| Feature vector Wavelet Dimension | 32x1 | 11x1 | 17x1 |

Table A.2: Feature vectors before and after applying PCA for 1 channel.

| 6 Channels | Without PCA | PCA (95%) | PCA (99%) |
|---|---|---|---|
| Feature Vector Fourier Dimension | 156x1 | 24x1 | 46x1 |
| Feature vector Wavelet Dimension | 192x1 | 60x1 | 78x1 |

Table A.3: Feature vectors before and after applying PCA for 6 channels.

## A.4 Classification

### A.4.1 SVM

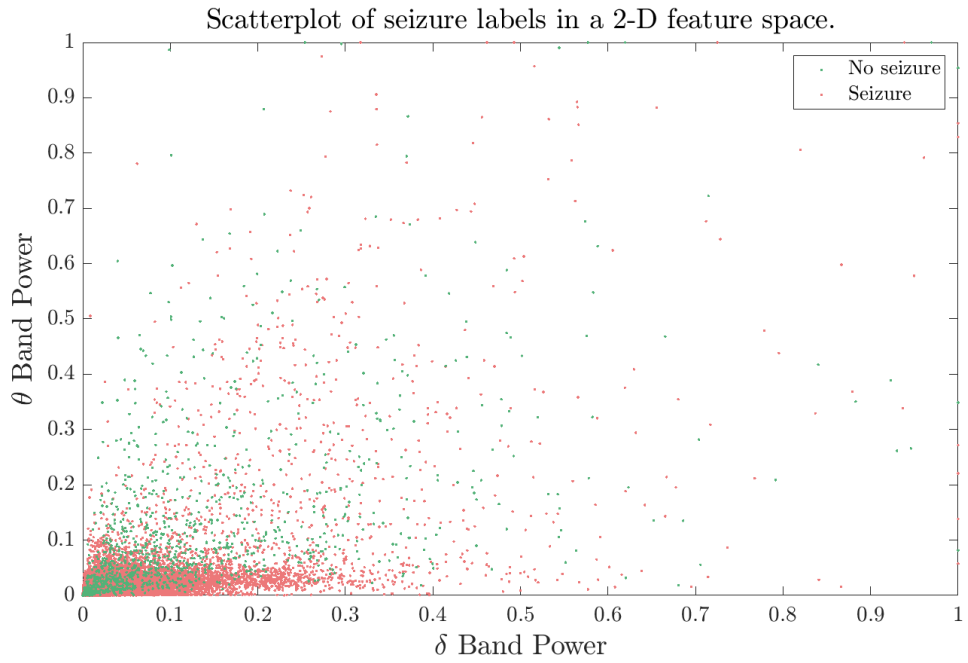



Figure A.5: Scatter plot showing 2-D feature space spanning the normalised $\delta$ and $\theta$ band power.

## A.5 Discussion of Results

### A.5.1 All Results

In all following graphs, each data point shown is the result obtained for a single patient. The patient that consistently has a 0% sensitivity over all models is patient 6. This patient has been excluded from the mean.

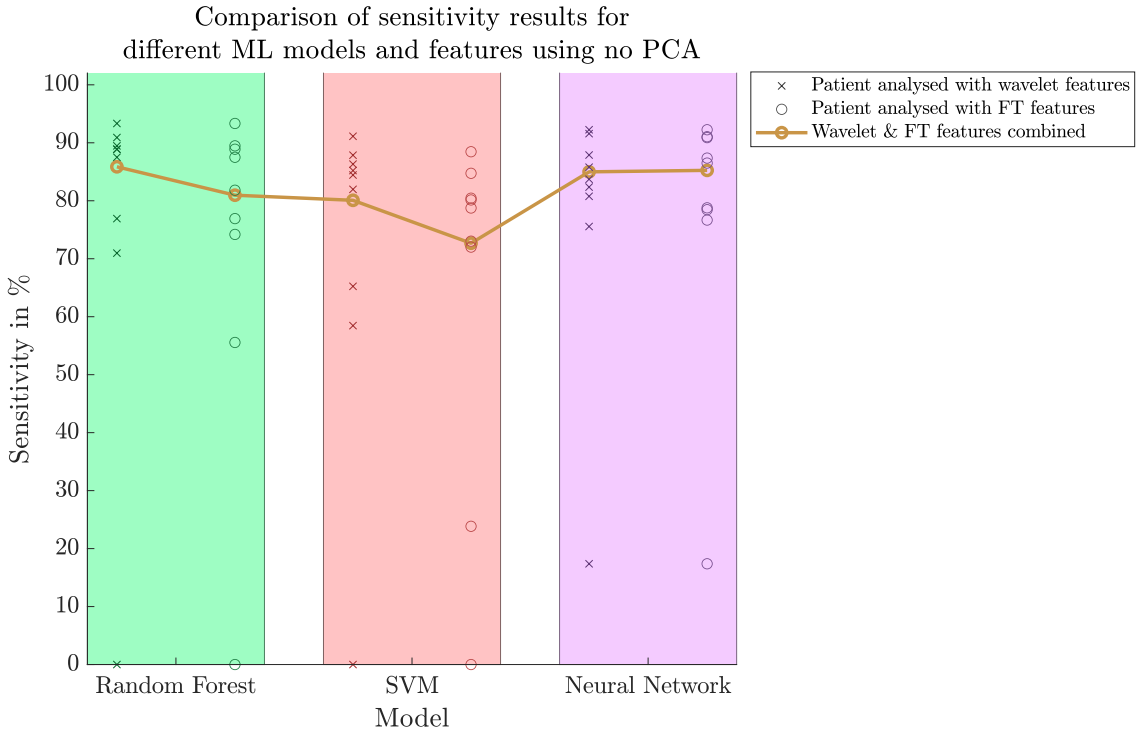



Figure A.6: Results of classification sensitivity for three Machine Learning (ML) models without using Principal Component Analysis (PCA). Each data point in the figure is the sensitivity result of a single patient.

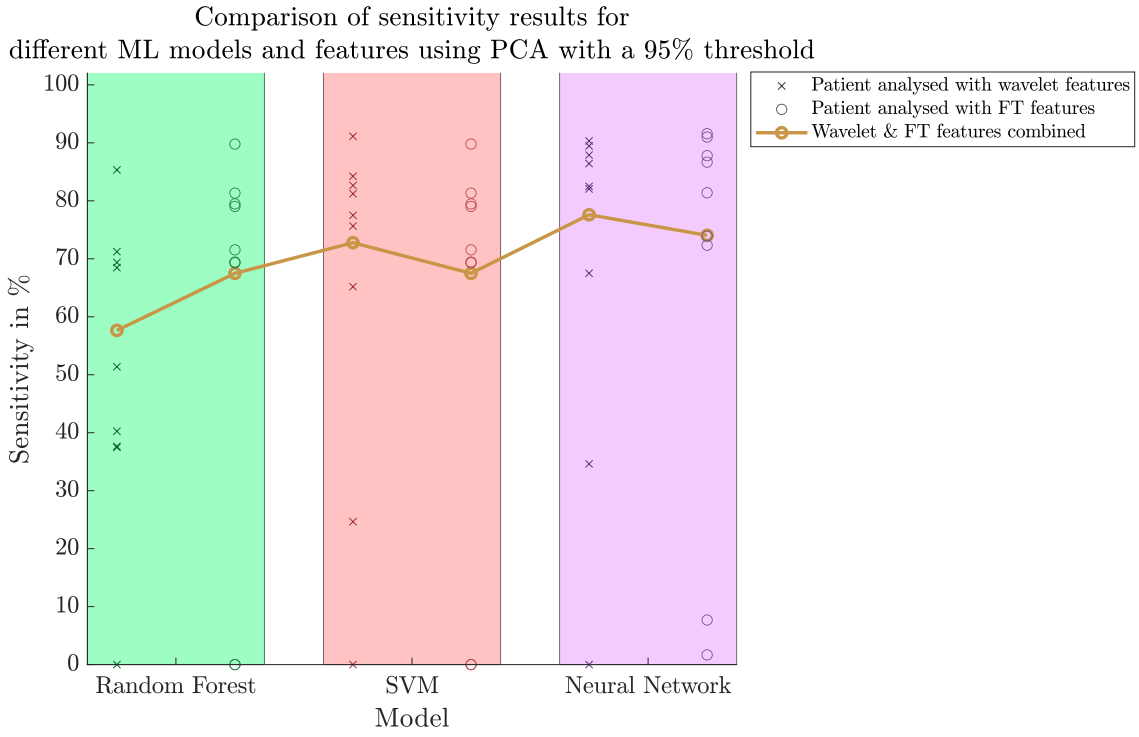



Figure A.7: Results of classification sensitivity for three ML models using 95% threshold for PCA. Each data point in the figure is the sensitivity result of a single patient.

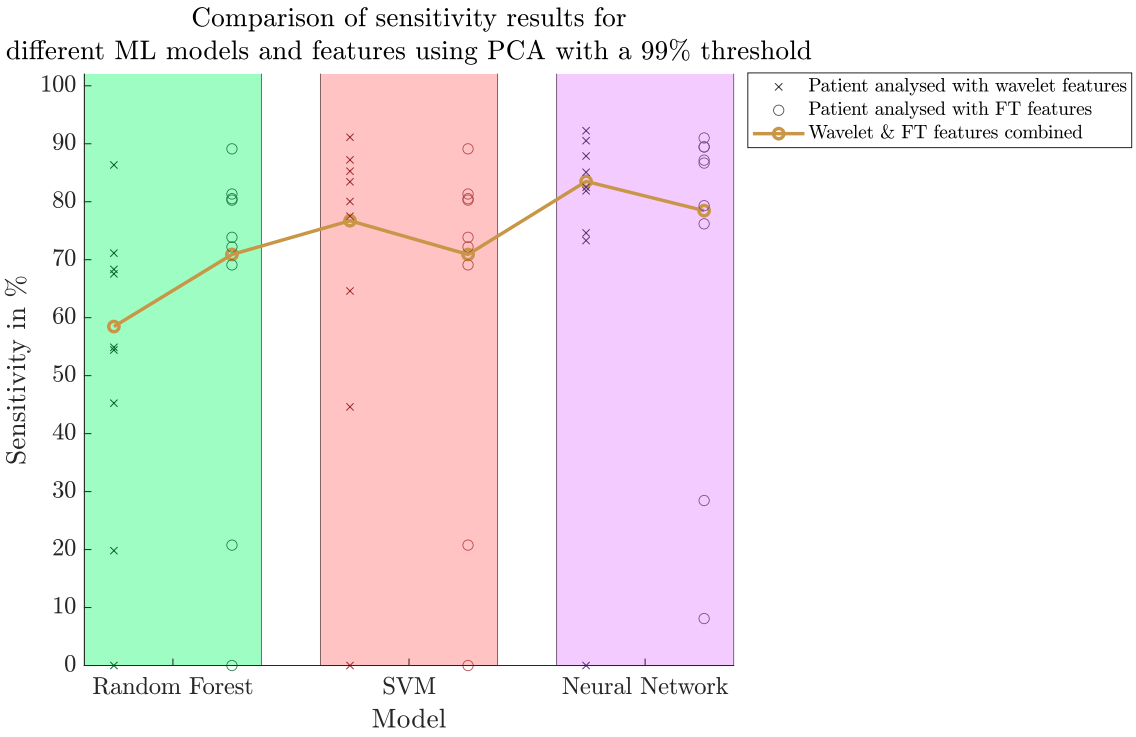



Figure A.8: Results of classification sensitivity for three ML models using 99% threshold for PCA. Each data point in the figure is the sensitivity result of a single patient.

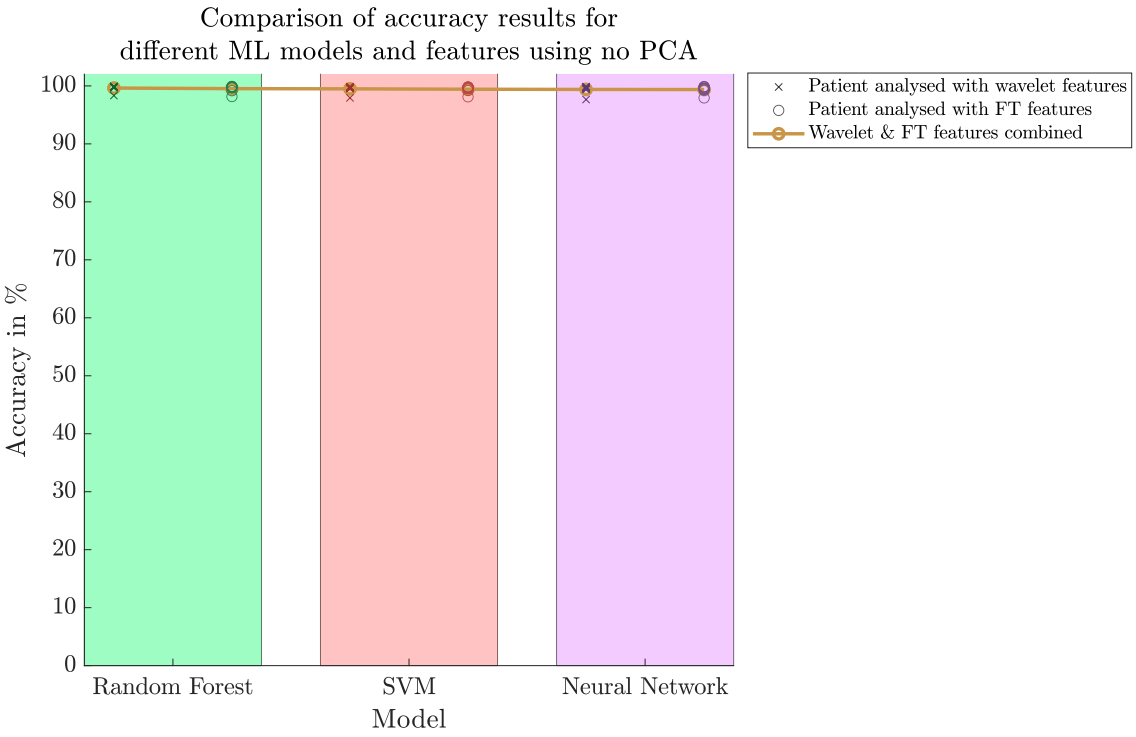



Figure A.9: Results of classification accuracy for three ML models without using PCA. Each data point in the figure is the accuracy result of a single patient.

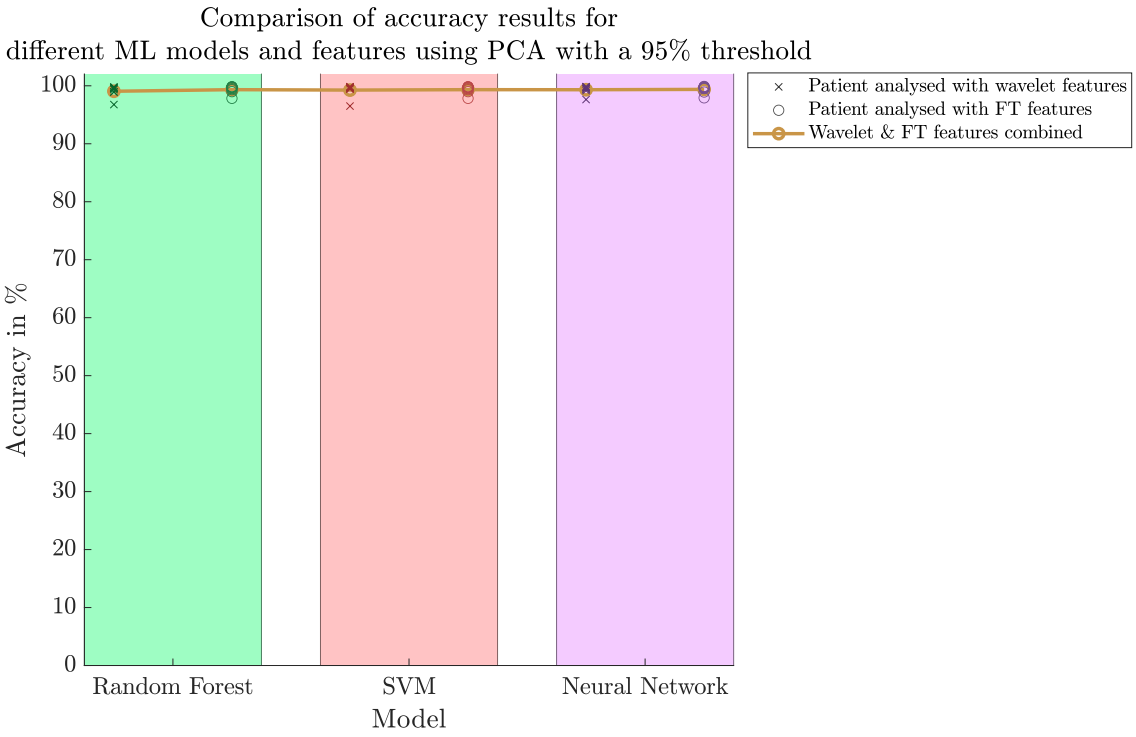



Figure A.10: Results of classification accuracy for three ML models using 95% threshold for PCA. Each data point in the figure is the accuracy result of a single patient.

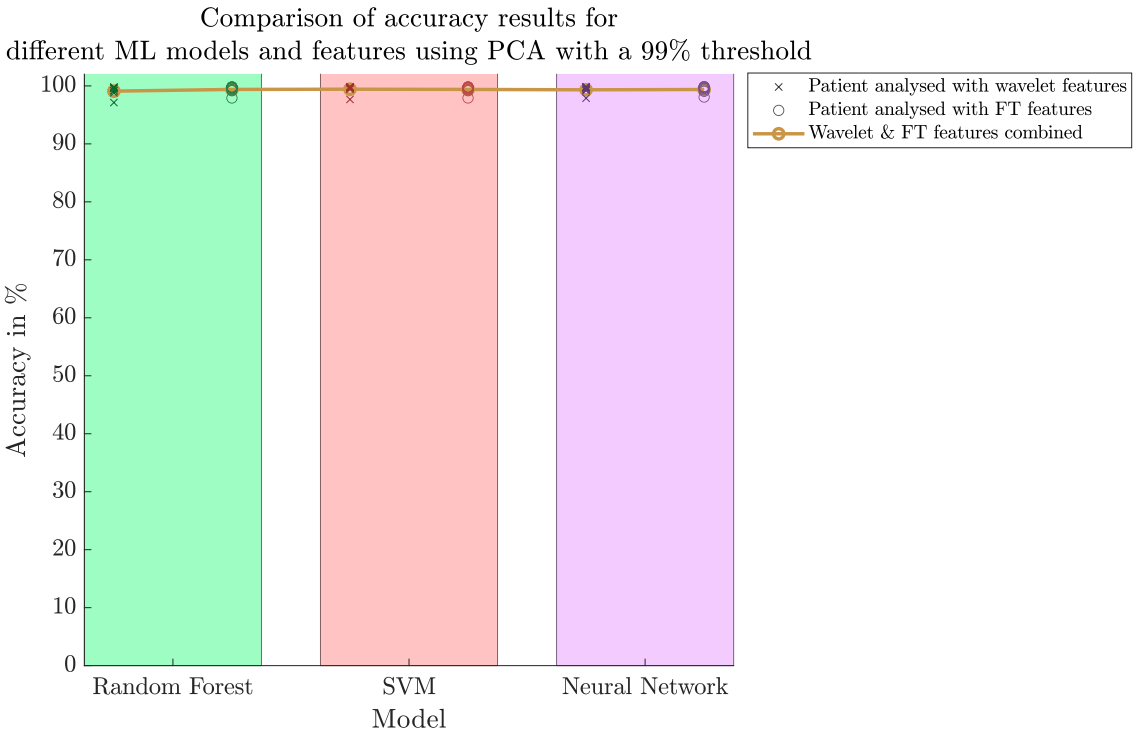



Figure A.11: Results of classification accuracy for three ML models using 95% threshold for PCA. Each data point in the figure is the accuracy result of a single patient.

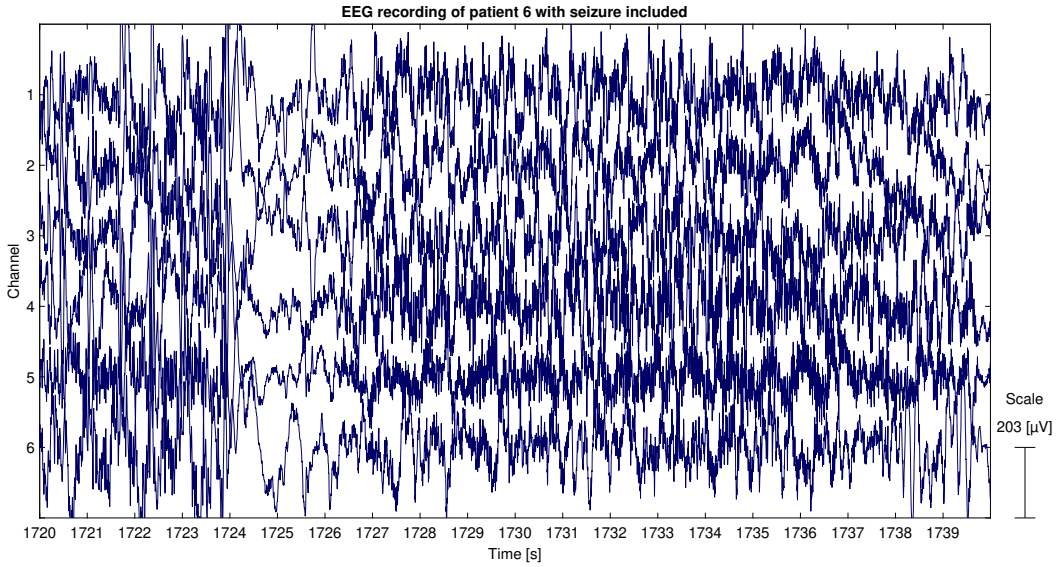



Figure A.12: A window showing the evaluated channels during a seizure for patient 6, the seizure starts at 1724 and ends at 1738 seconds.

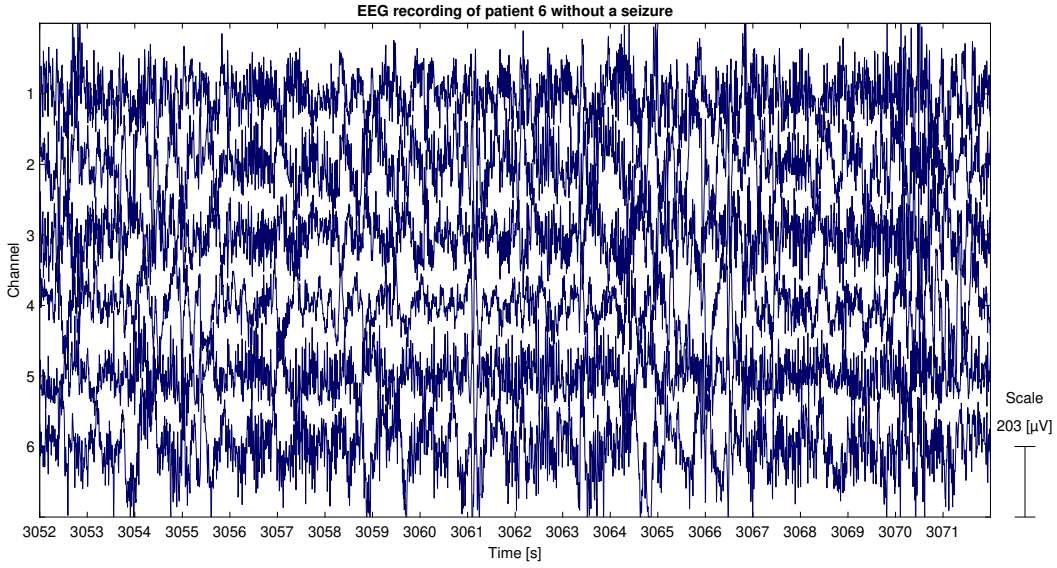



Figure A.13: A window showing the evaluated channels while no seizure is happening for patient 6. Note the similarity with the channel showing a seizure.

# Appendix B

# Training results

## B.1 Hyperparameter tuning results

| | Kernel Function | KernelScale | BoxConstraint |
|---|---|---|---|
| Per patient | rbf | 0.0177 | 0.0010 |
| Combined patient | rbf | 3.4575 | 211.68 |

Table B.1: Optimal hyperparameters after 50 iterations for the Support Vector Machine (SVM) model.

| | LayerSizes | Activations | Lambda |
|---|---|---|---|
| Per patient | 300 | none | $9.7118 * 10^{-5}$ |
| Combined patient | [289 12] | tanh | $2.4645 * 10^{-5}$ |

Table B.2: Optimal hyperparameters after 50 iterations for the Neural Network (NN) model.

## B.2 Results per patient

### B.2.1 Random Forests

| | 1 Channel | | 9 Channels | |
|---|---|---|---|---|
| Patient number | Accuracy | Sensitivity | Accuracy | Sensitivity |
| 01 | 99.50 | 73.33 | 99.85 | 93.33 |
| 03 | 99.17 | 55.56 | 99.17 | 55.56 |
| 04 | 99.30 | 38.46 | 99.77 | 76.92 |
| 05 | 99.33 | 78.95 | 99.67 | 89.47 |
| 06 | 99.81 | 0.00 | 99.81 | 0.00 |
| 07 | 98.89 | 0.00 | 99.82 | 81.82 |
| 08 | 98.33 | 67.74 | 98.17 | 74.19 |
| 09 | 99.65 | 66.67 | 99.91 | 88.89 |
| 10 | 99.88 | 87.50 | 99.88 | 87.50 |
| Average | 99.32 | 52.02 | 99.56 | 71.97 |

Table B.3: Results of random forest with 100 trees with fourier features and 6 channels using no PCA

| | 1 Channel | | 9 Channels | |
|---|---|---|---|---|
| Patient number | Accuracy | Sensitivity | Accuracy | Sensitivity |
| 01 | 99.25 | 73.33 | 99.87 | 93.33 |
| 03 | 99.38 | 66.67 | 99.79 | 88.89 |
| 04 | 99.22 | 38.46 | 99.77 | 76.92 |
| 05 | 99.33 | 78.95 | 99.67 | 89.47 |
| 06 | 99.81 | 0.00 | 99.81 | 0.00 |
| 07 | 99.35 | 45.45 | 99.91 | 90.91 |
| 08 | 97.33 | 51.61 | 98.33 | 70.97 |
| 09 | 99.65 | 66.67 | 99.83 | 88.89 |
| 10 | 99.94 | 93.75 | 99.88 | 87.50 |
| Average | 99.25 | 57.21 | 99.65 | 76.32 |

Table B.4: Results of random forest with 100 trees with wavelet features and 6 channels using no PCA

### B.2.2 Support Vector Machine

| | No PCA | | PCA(99) | | PCA(95) | |
|---|---|---|---|---|---|---|
| Patient | Accuracy | Sensitivity | Accuracy | Sensitivity | Accuracy | Sensitivity |
| 1 | 99.74 | 88.46 | 99.74 | 89.79 | 99.77 | 89.13 |
| 3 | 99.56 | 80.14 | 99.50 | 79.03 | 99.58 | 80.28 |
| 4 | 99.21 | 23.85 | 98.98 | 0.00 | 99.19 | 20.77 |
| 5 | 99.50 | 84.74 | 99.33 | 79.47 | 99.37 | 80.53 |
| 6 | 99.80 | 0.00 | 99.80 | 0.00 | 99.80 | 0.00 |
| 7 | 99.70 | 73.03 | 99.67 | 69.32 | 99.73 | 73.86 |
| 8 | 98.12 | 72.01 | 97.83 | 69.44 | 97.92 | 69.10 |
| 9 | 99.82 | 80.44 | 99.84 | 81.33 | 99.83 | 81.33 |
| 10 | 99.77 | 78.75 | 99.76 | 71.54 | 99.76 | 72.21 |
| Average | 99.47 | 64.60 | 99.38 | 59.99 | 99.44 | 63.02 |

Table B.5: Support Vector Machine results per patient with Fourier features and 6 channels. Values are in percentages.

| | No PCA | | PCA(99) | | PCA(95) | |
|---|---|---|---|---|---|---|
| Patient | Accuracy | Sensitivity | Accuracy | Sensitivity | Accuracy | Sensitivity |
| 1 | 99.72 | 91.13 | 99.70 | 91.13 | 99.70 | 91.13 |
| 3 | 99.73 | 87.86 | 99.77 | 81.22 | 99.77 | 87.22 |
| 4 | 99.58 | 58.46 | 99.43 | 24.65 | 99.43 | 44.62 |
| 5 | 99.53 | 86.32 | 99.48 | 82.65 | 99.50 | 85.26 |
| 6 | 99.80 | 0.00 | 99.80 | 0.00 | 99.80 | 0.00 |
| 7 | 99.77 | 81.97 | 99.60 | 77.50 | 99.72 | 77.50 |
| 8 | 97.93 | 65.24 | 96.50 | 65.19 | 97.67 | 64.61 |
| 9 | 99.81 | 84.44 | 99.84 | 84.23 | 99.84 | 83.44 |
| 10 | 99.86 | 85.21 | 99.76 | 75.64 | 99.82 | 80.04 |
| Average | 99.53 | 71.18 | 99.32 | 64.69 | 99.47 | 68.20 |

Table B.6: Support Vector Machine results per patient with wavelet features and 6 channels. Values are in percentages.

### B.2.3 Neural Network

| | No PCA | | PCA(99) | | PCA(95) | |
|---|---|---|---|---|---|---|
| Patient | Accuracy | Sensitivity | Accuracy | Sensitivity | Accuracy | Sensitivity |
| 1 | 99.35 | 75.29 | 99.31 | 71.46 | 99.26 | 71.50 |
| 3 | 99.23 | 69.72 | 98.08 | 3.61 | 98.90 | 54.44 |
| 4 | 99.05 | 22.31 | 98.94 | 0.00 | 99.05 | 16.15 |
| 5 | 99.22 | 81.58 | 99.07 | 75.26 | 99.10 | 76.84 |
| 6 | 99.80 | 0.00 | 99.80 | 0.00 | 99.80 | 0.00 |
| 7 | 98.99 | 30.45 | 99.01 | 17.20 | 98.94 | 18.94 |
| 8 | 97.50 | 64.96 | 96.75 | 47.88 | 97.28 | 60.45 |
| 9 | 99.71 | 72.22 | 99.35 | 30.22 | 99.58 | 64.11 |
| 10 | 99.89 | 90.92 | 99.87 | 87.75 | 99.85 | 86.50 |
| Average | 99.19 | 56.38 | 98.91 | 37.04 | 99.09 | 49.88 |

Table B.7: Neural Network results per patient with Fourier features and 1 channel. Values are in percentages.

| | No PCA | | PCA(99) | | PCA(95) | |
|---|---|---|---|---|---|---|
| Patient | Accuracy | Sensitivity | Accuracy | Sensitivity | Accuracy | Sensitivity |
| 1 | 99.50 | 86.50 | 99.67 | 87.79 | 99.57 | 87.17 |
| 3 | 99.40 | 87.36 | 99.52 | 81.39 | 99.65 | 89.44 |
| 4 | 99.63 | 78.46 | 98.92 | 7.69 | 99.07 | 28.46 |
| 5 | 99.25 | 91.05 | 99.67 | 91.58 | 99.32 | 89.47 |
| 6 | 99.81 | 17.38 | 99.80 | 1.67 | 99.80 | 8.10 |
| 7 | 99.54 | 76.67 | 99.61 | 73.94 | 99.64 | 79.32 |
| 8 | 97.93 | 78.78 | 97.92 | 72.34 | 98.08 | 76.19 |
| 9 | 99.83 | 90.89 | 99.84 | 86.67 | 99.84 | 86.67 |
| 10 | 99.89 | 92.25 | 99.89 | 91.00 | 99.88 | 91.00 |
| Average | 99.42 | 77.70 | 99.43 | 66.01 | 99.43 | 70.65 |

Table B.8: Neural Network results per patient with Fourier features and 6 channels. Values are in percentages.

| | No PCA | | PCA(99) | | PCA(95) | |
|---|---|---|---|---|---|---|
| Patient | Accuracy | Sensitivity | Accuracy | Sensitivity | Accuracy | Sensitivity |
| 1 | 99.29 | 72.08 | 99.22 | 68.17 | 99.20 | 67.54 |
| 3 | 99.73 | 84.31 | 99.40 | 69.31 | 99.38 | 71.67 |
| 4 | 99.06 | 20.77 | 98.98 | 6.92 | 99.02 | 9.23 |
| 5 | 99.18 | 80.00 | 99.12 | 75.79 | 99.07 | 76.32 |
| 6 | 99.80 | 0.00 | 99.80 | 0.00 | 99.80 | 0.00 |
| 7 | 99.27 | 46.82 | 99.04 | 25.23 | 99.09 | 35.08 |
| 8 | 97.23 | 56.60 | 96.93 | 53.38 | 96.88 | 53.05 |
| 9 | 99.73 | 77.44 | 99.31 | 33.33 | 99.75 | 79.56 |
| 10 | 99.89 | 89.71 | 99.90 | 90.33 | 99.89 | 90.29 |
| Average | 99.24 | 58.64 | 99.08 | 46.94 | 99.12 | 53.64 |

Table B.9: Neural Network results per patient with wavelet features and 1 channel. Values are in percentages.

| | No PCA | | PCA(99) | | PCA(95) | |
|---|---|---|---|---|---|---|
| Patient | Accuracy | Sensitivity | Accuracy | Sensitivity | Accuracy | Sensitivity |
| 1 | 99.45 | 85.79 | 99.51 | 86.42 | 99.32 | 85.08 |
| 3 | 99.40 | 82.36 | 99.46 | 82.50 | 99.21 | 82.50 |
| 4 | 99.69 | 80.77 | 99.17 | 34.62 | 99.61 | 74.62 |
| 5 | 99.52 | 91.58 | 99.42 | 89.47 | 99.33 | 90.53 |
| 6 | 99.79 | 17.38 | 99.80 | 0.00 | 99.79 | 0.00 |
| 7 | 99.71 | 83.71 | 99.71 | 82.05 | 99.59 | 81.89 |
| 8 | 97.65 | 75.56 | 97.63 | 67.50 | 97.88 | 73.31 |
| 9 | 99.82 | 87.89 | 99.85 | 87.89 | 99.82 | 87.89 |
| 10 | 99.89 | 92.25 | 99.81 | 90.33 | 99.86 | 92.25 |
| Average | 99.44 | 77.48 | 99.37 | 68.97 | 99.38 | 74.23 |

Table B.10: Neural Network results per patient with wavelet features and 6 channels. Values are in percentages.