Absence seizure prediction using recurrent neural networks

Corniël Joosse

CE-ES-2018-15

Delft University of Technology

Abstract

Absence seizures have a real-life impact on epileptic subjects, as day-to-day tasks can by suddenly interrupted making for dangerous situations. Though a lot of work has been done on seizure detection, to limit the impact on epileptic patients, the true necessity lies in timely prediction of seizures before they manifest. Various attempts have been made using conventional algorithms to accurately predict seizures however, so far, results are not that encouraging. In this work, we applied various machine-learning algorithms, in an attempt to identify complex, multi-dimensional epileptic precursors in brain recordings. Three types of neural networks are used in this feasibility study, namely Multi-Layer Perceptron (MLP) networks, Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks. The used input data was annotated Electrocorticography (ECoG) data, recorded in living mutant rodents, containing epileptic events at an interval of about one minute. The data was pre-processed for better learning performance by data normalisation and by generating distinctive training features. The neural networks were configured as three-class classifiers, distinguishing among inter-ictal, pre-ictal and ictal periods. A grid-search approach was applied to determine the best set of parameters for the neural networks. Despite our best efforts, the relation between the input data and output data could not be learned in a reliable way. The maximum reached Average Prediction Rate (APR) was 0.57 with a prediction time of 3.1s when using the normalised data as input and 0.65 with a prediction time of 6.1s when using the distinctive features as input. These results essentially signify good detection but virtually no prediction of upcoming seizures. The evaluation of the experimental findings has revealed that the employed ECoG recordings were ill-selected for training our various neural-network models. Also, a non-conclusive exploratory experiment is performed by applying a Weibull Time-To-Event Recurrent Neural Network (WTTE-RNN) on a sub-set of the normalised input data. The experiment has yielded some positive results, a short-notice prediction of the upcoming seizure in some cases, encouraging for further exploration of this approach. Despite the limited success of this work, however, through its extended forensics analysis, it has paved the crucial, initial steps in the direction of seizure prediction.

TU Delft Delft University of Technology

CE Lab Computer Engineering Laboratory

Absence seizure prediction using recurrent neural networks

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

Corniël Joosse
born in Mariekerke, Netherlands

Computer Engineering
Department of Electrical Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Absence seizure prediction using recurrent neural networks

by Corniël Joosse

Abstract

Absence seizures have a real-life impact on epileptic subjects, as day-to-day tasks can by suddenly interrupted making for dangerous situations. Though a lot of work has been done on seizure detection, to limit the impact on epileptic patients, the true necessity lies in timely prediction of seizures before they manifest. Various attempts have been made using conventional algorithms to accurately predict seizures however, so far, results are not that encouraging. In this work, we applied various machine-learning algorithms, in an attempt to identify complex, multi-dimensional epileptic precursors in brain recordings. Three types of neural networks are used in this feasibility study, namely MLP networks, LSTM networks and GRU networks. The used input data was annotated ECoG data, recorded in living mutant rodents, containing epileptic events at an interval of about one minute. The data was pre-processed for better learning performance by data normalisation and by generating distinctive training features. The neural networks were configured as three-class classifiers, distinguishing among inter-ictal, pre-ictal and ictal periods. A grid-search approach was applied to determine the best set of parameters for the neural networks. Despite our best efforts, the relation between the input data and output data could not be learned in a reliable way. The maximum reached APR was 0.57 with a prediction time of 3.1s when using the normalised data as input and 0.65 with a prediction time of 6.1s when using the distinctive features as input. These results essentially signify good detection but virtually no prediction of upcoming seizures. The evaluation of the experimental findings has revealed that the employed ECoG recordings were ill-selected for training our various neural-network models. Also, a non-conclusive exploratory experiment is performed by applying a WTTE-RNN on a sub-set of the normalised input data. The experiment has yielded some positive results, a short-notice prediction of the upcoming seizure in some cases, encouraging for further exploration of this approach. Despite the limited success of this work, however, through its extended forensics analysis, it has paved the crucial, initial steps in the direction of seizure prediction.

Laboratory	:	Computer Engineering
Codenumber	:	CE-MS-2018-15

Committee Members :

Advisor:	Dr. ir. C. Strydis, Neuroscience dep., Erasmus MC
Chairperson:	Prof. dr. ir. W.A. Serdijn, BE, TU Delft
Member:	Dr. ir. B. Hunyadi, CAS, TU Delft
Member:	Prof. dr. ir. G.N. Gaydadjiev, DS, TU Delft

Dedicated to the boy with the pearls who kept believing in me.

Contents

List of Figures

List of Tables

List of Acronyms

EEG Electroencephalography

ECoG Electrocorticography

EMG Electromyography

EMC Erasmus Medical Center

SWD Spike-and-Wave Discharge

TAS Typical Absence Seizure

CAE Childhood Absence Epilepsy

JAE Juvenile Absence Epilepsy

ANN Artificial Neural Network

MLP Multi-Layer Perceptron

RNN Recurrent Neural Network

LSTM Long Short-Term Memory

GRU Gated Recurrent Unit

BPNN Back-propagation Neural Network

BPTT Backpropagation Through Time

WBAN Wireless Body Area Network

SVM Support Vector Machine

ApEn Approximate Entropy

DWT Discrete-Wavelet Transform

DB4 Daubechies 4

WT Wavelet Transform

AR model Autoregressive Model

FFT Fast Fourier Transform

LR Logistic Regression

OI Order Index

PCA Principle Component Analysis

MVI Multiscale Variance Index

DWNN Dynamic Wavelet Neural Network

ANFIS Adaptive Neuro-fuzzy Inference System

NMSE Normalised Mean Squared Error

SSIM Structural Similarity)

DT Decision Tree

DA Discriminant Analysis

KNN K-nearest Neighbor

KF Kalman Filter

DT Decision Tree

LLS Linear Least Squares

LDA Linear Discriminant Analysis

FPR False Positive Rate

FDR False Detection Rate

TPR True Positive Rate

TNR True Negative Rate

ADR Average Detection Rate

APR Average Prediction Rate

TP true-positive

FP false-positive

TN true-negative

FN false-negative

ReLu Rectified Linear units

LReLu Leaky ReLu

TTE Time-To-Event

WTTE Weibull Time-To-Event

WTTE-RNN Weibull Time-To-Event Recurrent Neural Network

Acknowledgements

Along the journey to get to this thesis, that took a little long, I was helped and supervised by Christos Strydis of the Neuroscience department of the Erasmus Medical Centre. I want to thank him for for his advice and ideas during the project and also his endless patience in supervising me. His reviews of the work were very valuable. I also want to thank Jan-Harm Betting and prof. Georgi Gaydadjiev, which later in the project thoroughly reviewed my work for which I'm very grateful. The work of Wouter Serdijn also has been unmissable, as he took on the role of supervising professor quite late in the project. Another word of thanks I want to give to Abel Sensors, for giving me the opportunity to still work on this thesis next to my job over there. Last but not least, I want to thank family and friends who supported me throughout the whole project, and especially Martijn who kept supporting and motivating me.

Corniël Joosse
Delft, The Netherlands
May 22, 2020

1 Introduction

The brain is an important organ as it controls most of the body's behaviour. Disruptions in the brain can therefore have a high effect on the body. Absence seizures are one example of such disruptions, and they highly influence the subject as it blocks the normal brain operation totally. An epileptic seizure is a transient occurrence of signs and/or symptoms due to abnormal excessive or synchronous neuronal activity in the brain [1]. We can differentiate among different types of seizures, one of which are absence seizures. An absence seizure lasts for two or three seconds up to about 20 seconds and ends suddenly with resumption of the pre-absence activity, as if it had not been interrupted. It sometimes only manifests with impairment of consciousness, but is often accompanied by mild clonic jerks, relaxation or contraction of muscles, or common automatisms like swallowing [2]. Absence seizures are more common in children than in adults as the seizures can be the result of a syndrome called Childhood Absence Epilepsy. There are multiple comorbidities of Childhood Absence Epilepsy known, showing the impact on a child's life, including attentional problems, anxiety, depression, social isolation, and low self-esteem [3].

Seizure detection and prediction are methods which try to classify a certain amount of measurements based on previously set rules or learned models. The most common type of measurements used are brain-signal recordings, either on or under the skull, resulting in Electroencephalography (EEG) and Electrocorticography (ECoG) measurements, respectively. The brain measurements are then classified using machine-learning algorithms, like Support Vector Machines (SVMs) or neural-networks. Despite the amount of research done on predicting seizures so far, the results are not that encouraging. Part of the reason is that the prediction time is not high enough to be able to react on the prediction, or the accuracy is too low for reliable usage. In this work we look at applying other types of machine-learning algorithms to brain-signal recordings, in order to verify the feasibility of absence-seizure prediction with higher accuracy and with longer prediction times.

1.1 Motivation and problem statement

While the first research on seizure prediction dates from the 20th century, most research has been done more recently. Several machine-learning algorithms have been applied to EEG and ECoG measurements in previous research, but they only focused on non-recurrent machine-learning algorithms. In this work, on the contrary, we are going to use Recurrent Neural Networks (RNNs) to predict absence seizures.

A machine-learning algorithm can be either recurrent or non-recurrent. A recurrent algorithm uses the output of the previous iteration as input in the next iteration. By doing so, it is possible to base the output of the current iteration not only on the input

of the current time-step, but also on previous ones. In other words, these are systems with memory. This leads to the possibility to learn from markers in the data that build up gradually over time. How the recurrency is managed internally differs for each recurrent algorithm. Since seizures are recurring events in a patient and supposedly have a trigger which builds up gradually, it is our hope that by using RNNs, we will be able to increasingly improve our predictions of future events based on learned past events.

At the Erasmus MC Neuroscience department, previous research has been done on ECoG measurements of mice [4]. The data used in [4] is also used in this work. It contains high-quality annotations of when seizures start and end. This makes the data suitable for use with supervised machine-learning, a method we are using in this thesis. The availability of such a data-set is a unique possibility and enabling asset needed for research as such done in this work.

The work is further motivated by the large social impact to be had from reliable seizure prediction. If the prediction on rodent brains can also be applied to humans, a patient can know that a seizure is coming. In effect, the patient has a chance to act upon the prediction, or an automated system can be used to suppress the seizure e.g. by using deep brain stimulation.

Generally speaking, there is a hope that the more diverse yet relevant bio-data we can combine, the better we will be able to predict seizures in the future. This could be facilitated by a Wireless Body Area Network (WBAN): a system where multiple sensors and actuators on the human body can communicate wirelessly with each other. Sensors on different parts of the body, providing input to a seizure prediction system can be part of such a WBAN, as can the system that suppresses or prevents the seizure from happening. To this end, we have conducted a survey on available WBAN systems and communication protocols, which is appended to this thesis in Appendix A.

1.2 Thesis scope and contributions

In this thesis, a number of different neural-network algorithms will be applied to ECoG measurements of rodents to verify the feasibility of predicting absence seizures. Because we do not know which features and parameters will lead to the best result, we will apply a grid-search approach on two different data sets. The first data-set consists of normalised input data and the second data-set are distinctive generated features based on the normalised input data-set. The results will be verified based on Average Prediction Rate (APR), True Positive Rate (TPR), True Negative Rate (TNR), prediction time and neural-network loss. An exploratory experiment will also be performed to evaluate the feasibility of applying a Weibull Time-To-Event Recurrent Neural Network (WTTE-RNN) to a sub-set of the input data. This in order to advise follow-up research whether this is a promising path to follow.

1.2.1 Thesis goal

The main goal of the thesis can be formulated as follows:

Determine the best-performing set of input data, neural-network type and parameters for predicting absence seizures.

The first step towards achieving this goal is the selection of the appropriate features which can be used to form the input data-sets. For this, we will describe the background of some features and determine which data-sets to use as input for the neural networks. The second step is to determine the types of neural networks and the parameters to use within the grid-search approach. The third step is to evaluate the results of the experiments and select the best-performing set of data, neural-network type and parameters.

1.2.2 Thesis contributions

The following contributions were made in this thesis:

- RNNs were applied on ECoG recordings, using a grid-search approach to compare multiple combinations of neural-network parameters, in order to predict absence seizures.
- Research has been done on which features can be generated from ECoG data, to be able to show distinctive characteristics of ECoG measurements.
- A guideline is provided for future research on machine-learning based seizure-prevention approaches, by setting up an extensive experimental setup and by providing a thorough analysis of working approaches and limitations.
- A regression method was used to predict a Time-To-Event (TTE) value depicting the time to the next seizure, using Weibull distribution parameters, in combination with a RNN.

1.3 Thesis organisation

This thesis consists of the following parts. Firstly, background information which is needed to understand the research is given in chapter 2. In chapter 3, related research is discussed. The design and implementation of the machine-learning algorithms is set out in chapter 4. The evaluation of the experiments is described in chapter 5 and the results are discussed in chapter 6.

2 Background

In this chapter, background information is given on several topics, needed for better understanding of the rest of this thesis. In section 2.1 we will describe what seizures are and what their main properties are specifically for the case of absence seizures. In section 5.1, more information is given on the used data recordings. Some relevant signal processing algorithms are set-out in section 2.2. machine-learning is explained in a general way in section 2.3 while specific types of machine-learning are explained in the sections that follow (section 2.4 and section 2.5).

2.1 Absence seizures

In this section we will look at what brain signals can be defined, what absence seizures are and how they manifest in the brain.

The brain consists of neurons that communicate with each other using electrical pulses and chemical signals. Each electrical pulse induces a little electromagnetic field, which is measurable using electrodes on the brain or on the scalp. Measuring brain signals on the scalp is what we call Electroencephalography (EEG) and measuring underneath the scalp, on the brain itself, is called Electrocorticography (ECoG). The following frequency bands are defined as part of the recorded electrical brain signals and are commonly used in the literature:

- Delta (δ) band: < 4 Hz
- Theta (θ) band: 4.5-8 Hz
- Alpha (α) band: 8.5-12 Hz
- Beta (β) band: 12.5-36 Hz
- Gamma (γ) band: 36.5-100 Hz

The exact cutoff frequencies can differ from publication to publication and also depend on the used algorithm to filter or extract bands.

The general definition of an epileptic seizure is given by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE) as: a transient occurrence of signs and/or symptoms due to abnormal excessive or synchronous neuronal activity in the brain [1]. Absence seizures, also called absences, are a specific type of seizures, which we can characterise clinically and/or based on brain measurements. In this thesis we focus on Typical Absence Seizures (TASs) because they are the most common type of absence seizures and the expertise of the department of neuroscience at the Erasmus Medical Center (EMC) for a number of years.

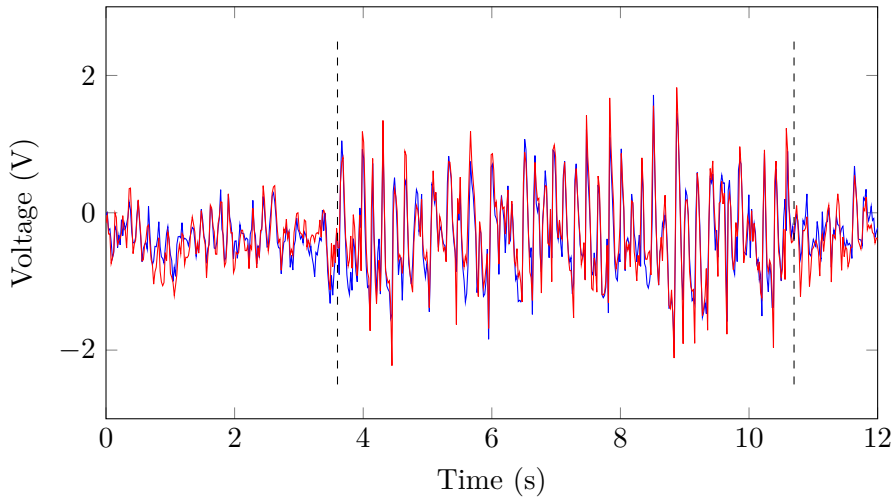


Figure 2.1: A 2-channel ECoG signal containing a seizure between the vertical dashed lines, visible by typical SWDs in the signal.

Clinically, we can identify a TAS by a brief and sudden loss of consciousness, which generally also results in the interruption of the ongoing activity of the patient. An absence lasts usually for a few to about 20 seconds and ends abruptly with the resumption of the previous activity, as if no seizure occurred. Although some absences only result in the loss of consciousness, often they are accompanied by (in the case of humans) mild jerks of eyelids or mouth, drooping of the head, relaxation or contraction of limb muscles, automatisms as lip licking or swallowing, sweating and dilation of pupils [2].

EEG or ECoG recordings show the characteristics of the ictal period of an absence. Although EEG is measured on the scalp, and ECoG directly on the brain itself, they show similar characteristics. The advantages of ECoG measurements over EEG measurements are however, that the sensitivity of the recordings is better which results in a better signal-to-noise ratio, the placement of the electrodes is more flexible and and the recordings are more local. A typical absence seizure manifests generalised Spike-and-Wave Discharges (SWDs) with a frequency of typically 3-4.5 Hz, lasting for more than 3 seconds [2], see Figure 2.1.

Severe absence seizures occur more often in children than adults, and complex absence seizures are more common than simple absences in children, while simple absences are more common in adults. A simple absence means that the transient consciousness impairment is the only symptom, while complex absence seizures are combined with other manifestations, like muscular contractions or common automatisms [5]. Absence seizures are the hallmark seizure type in two epilepsy syndromes: Childhood Absence Epilepsy (CAE) and Juvenile Absence Epilepsy (JAE). Research has shown that absence seizures also have impact on one's life socially. From children with CAE we know that there are multiple comorbidities, including attentional problems, anxiety, depression, social isolation, and low self-esteem [3].

Another type of seizures to which often is referred, is the tonic-clonic seizure, also known as grand mal seizure. Clinically it is characterised by two phases. First the

patient loses consciousness while all the muscles stiffen, the tonic phase. Then, during the clonic phase, rapid jerking of the limbs occur. Looking at brain signals, the main difference is visible in how the SWDs manifest.

2.2 Signal processing

2.2.1 Discrete-Wavelet Transform (DWT)

Contrary to the Fourier Transform which loses all time-domain information, the Discrete-Wavelet Transform (DWT) transforms a signal in both frequency- and time-domain.

The Continuous Wavelet transform of a signal is the integral of the signal multiplied by scaled and shifted versions of a wavelet function. The DWT is a discrete version of this transform. It transforms a signal into several windows of varying length each holding frequency information. The size of the window determines the coarseness of the time information: a short window has a higher time-domain resolution and is used for high frequency information while a long window is used to get a finer low frequency resolution with a lower time-domain resolution.

To determine the DWT coefficients the signal is filtered using a high-pass and a low-pass filter. The low-pass filter filters out the lower half of the frequencies which result in the approximation coefficients. The high-pass filter filters out the upper half of the frequencies containing the level-1 detail coefficients. As half of the frequencies have been removed, we can remove half of the samples according to Nyquist's rule, depicted by the $\downarrow 2$ step.

This process can be repeated on the remaining signal of the low-pass filter, filtering out again the upper half of the remaining frequencies, resulting in the level-2 detail coefficients, and the lower half of the remaining frequencies, resulting in the approximation coefficients. Each time this process is applied, the detail and approximation coefficients double in frequency resolution and half in time resolution, see Figure 2.2. Also the cut-off frequency of the filter halves.

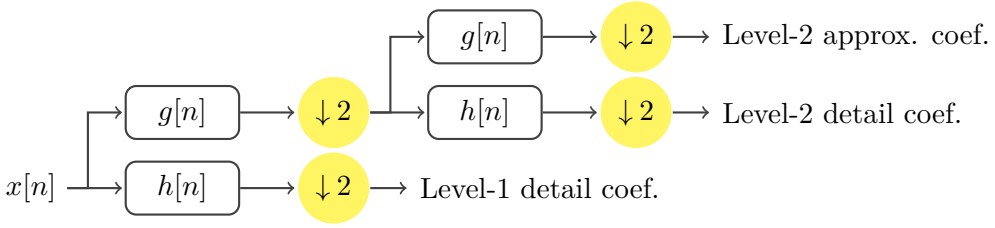


Figure 2.2: Block diagram of 2-level DWT signal decomposition

Because the time and frequency information both remain in the data, it is possible to do a reverse DWT which reconstructs the signal again. This allows for removing the coefficients of certain levels which removes the corresponding frequency range from the signal. The resulting signal is a filtered version of the input data in which only the selected frequencies remain.

2.2.2 Approximate Entropy (ApEn)

Approximate Entropy (ApEn) is a signal complexity measure developed with relatively short (starting at 100 samples) and noisy time-series in mind. These short time-series are often encountered in biological data-sets, as e.g.„ EEG [6], **ECG!** (**ECG!**) or endocrine hormone secretion measurements [7][8].

The ApEn algorithm has two input parameters, m and r which must be fixed to compute the ApEn. m represents the length of the window of data to use in each run and r specifies a filtering level. Calculating ApEn consists of the following simplified steps:

1. The input data x is divided into vector sequences $X_v(i)$ of length m, so that $X_v(i) = \{x(i), x(i+1), \cdots, x(i+m-1)\}$.

2. The distance between two vector sequences $X_v(i)$ and $X_v(j)$ is defined using the function $d(X_v(i), X_v(j))$.

3. For each $X_v(i)$ the number of vectors N is calculated such that $d(X_v(i), X_v(j)) \leq r$, this results in the estimated parameter c.

4. Then Φ is defined as mean value of the sum of $ln(c_i)$, where c_i is calculated for $m+1$ vectors.

5. The approximate entropy is finally calculated using the difference of Φ^m and Φ^{m+1}.

This results in an entropy which measures the logarithmic likelihood that patterns within r for m contiguous runs, remain close. For ECoG data this means that a low ApEn indicates a signal with repetitive patterns and a high ApEn indicates that the analysed signal contains non-similar patterns. See [7] and [6] for a more extensive explanation of the algorithm.

2.3 Machine-learning

The ability to learn is only exclusively available to living organisms, and especially human beings are very good at learning. With the invention of machine-learning we can also make a machine learn a relation between its input and output data. This can be done using supervised or unsupervised learning, where for supervised learning, the output is known for a set of training data. For unsupervised learning, the output is not known, but classified by the learning algorithm. Because our training data is annotated, we know the output and therefore use supervised learning, which is easier to handle than unsupervised learning. Unsupervised learning is in many ways still an unsolved problem.

We can differentiate between two types of machine-learning algorithms: regression and classification. Regression is targeted at predicting continuous values e.g., future parts of a signal. Classification algorithms aim at classifying a given input set among two or more distinct classes and have a discrete output where each output corresponds to a certain class.

There are multiple learning algorithms, of which some are simple and straightforward needing only a few data-points, and others are complex and require a lot of data to train on. One of the simplest regression algorithms is linear regression, where we learn the a and b parameter for the equation $y = ax + b$ using a set of training data-points.

The input and output data have to be preprocessed by extracting one or multiple features which describe the input data in such a way that it can be processed by the learning algorithm and minimises the amount of data the algorithm has to process. Specific machine-learning algorithms are discussed in the following sections.

2.3.1 Features and Feature extraction

A feature can be described as a characteristic or attribute of an object [9]. For example when looking at an image, features could be the amount of pixels, a histogram of the intensity, all red pixels or a subsample of all pixels. When looking at time series we can think of other features, e.g., the mean or standard deviation of a signal or the DWT coefficients of a certain level. Which features are the best suited ones to use is dependent on the learning algorithm, input and output data. In other words, what data is available and what the learning algorithm should learn. Important is that a feature describes a signal in a consistent and descriptive way so that a similar pattern can be described by the same or similar features, but a different signal will yield distinctive features. The process of generating features from the raw data is called feature extraction.

2.3.2 Training and testing

After the input data is converted into features, they are used to train the learning algorithm. Depending on the type of learning algorithm, a classification or regression model has to be trained. A regression model is trained to predict a numeric value as output while a classification model is trained to determine a class-value based on the input data. Classification can either be binary, resulting in two classes, or multi-class which can be seen as multiple binary classifications stacked on each other.

During training, the algorithm is exposed to samples containing a set of features and a corresponding output. The algorithm then updates its internal state in such a way that the error between the provided output and the self predicted output, based on the input, is minimised.

After all training samples have been processed by the learning algorithm one or multiple times, another set of unique samples is used to test the model. This time the internal state of the learning algorithm is not changed, but only the error between the predicted and real output is recorded. The average of all errors then indicates how well the model is trained.

2.4 Support Vector Machine (SVM)

An SVM is a machine-learning algorithm and binary classifier. The input data is first mapped onto a high-dimension feature space so it can handle more complex input-sample relations. Based on the samples in the feature space, an SVM tries to find a hyperplane

which divides the data points into two categories or classes. This hyperplane can then be used to classify new data into one of the two classes. The dividing hyperplane can be linear because of the high-dimension feature space [10].

When the amounts of each different output class are not evenly distributed, a dataset is called unbalanced. This can influence the learning process and result in a higher probability that a more frequent class is predicted. An optimised version for unbalanced datasets, the cost-sensitive SVM [11], keeps in consideration the class-balance of the output classes and assigns different weights to the input values when training the SVM. In this way, classification performs better because the less common class will also be classified correctly instead of being drowned under the more common class. Another variation of the binary SVM is the multi-class SVM [12] which can classify more than two classes. Essentially, this approach is a combination of multiple binary SVMs where the multi-class classification problem is divided into multiple two-class problems.

2.5 Artificial Neural Networks (ANNs)

A neural network is a more advanced machine-learning algorithm than the previously mentioned algorithms, and is loosely based on how a brain works. The input data is fed into a network of simple neuron models (simply called neurons henceforth), which all have a certain input weight and activation mechanism. The output of the neuron is in turn the input to the next layer of neurons. The way neurons are interconnected in a network, the number of neurons in each layer and the activation function of a neuron all highly influence the properties of the network.

2.5.1 Activation functions

In each neuron or gated unit of an ANN an activation function is used. It defines the output based on the summed input. Without an activation function, the output would explode very fast, especially because multiple neurons are chained together in ANNs. The output function normalises the output to a usable range for the next neuron, and offers non-linearity which allows a ANN to learn non-trivial and non-linear problems. The following activation functions are commonly used by ANNs.

The *tanh* activation function is simply the *tanh*-function applied to the summed input, see Equation 2.1. The function compresses the output into the range $[-1, 1]$, and has two asymptotes at -1 and 1, see Figure 2.3a.

$$f(x) = tanh(x) \tag{2.1}$$

The sigmoid (or logistic) activation function is defined as shown in Equation 2.2 and has an output between 0 and 1 with also two asymptotes.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

The softmax function is a normalised version of the sigmoid function for multiple output values which are normalised using the formula shown in Equation 2.3. The output of a neuron is not one value but a vector of values, called x. The output is

then normalised by applying the standard exponential function to every value in x, after which it is divided by the sum of all the exponentials of the values in x. By doing so, the cumulative output of a softmax output layer is always 1, which makes it suitable for multi-class classification. Each value then depicts the probability-score of each class.

$$f(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.3}$$

The Rectified Linear units (ReLu) function is a modified version of a linear function. It has an output of 0 for all negative input values, and the identity function for all positive input values, see Equation 2.4 (with $\alpha = 0$). This activation function however, can suffer from the dying ReLu problem. A ReLu neuron is called dead when the output of the neuron is always the same, independent of the input. Because of the learned weights on the inputs of the neuron, the summed input will never come above 0, so the output will also be 0. A neuron cannot come out of this state because the gradient on which the weights are updated is also 0.

There are variations of this function which solve the dying ReLu problem. The idea is the same as with the ReLu function. For values above zero, the activation function is the identity function, but below zero, the output is a fraction of the input: $y = \alpha \cdot x$ These ReLu variations use a non-zero value for α in Equation 2.4 where $0 < \alpha < 1$. For example, in the Parameteric ReLU (PReLu) α is a parameter that is being learned while training the network and the Leaky ReLu (LReLu) has α as input parameter.

$$f(\alpha, x) = \begin{cases} \alpha x, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases} \tag{2.4}$$

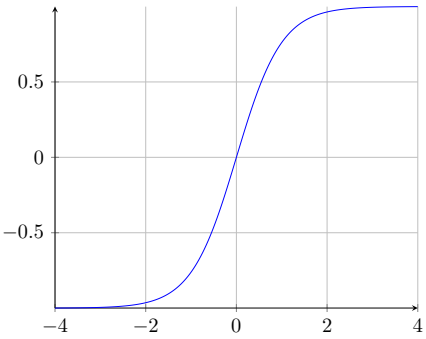
The SoftPlus activation function, also known as Smooth ReLu, is another activation function which prevents the dying ReLu problem. The SoftPlus activation function solves this not only by having a non-zero output for negative values, but also for an input of zero.

$$f(x) = ln(1 + e^x) \tag{2.5}$$

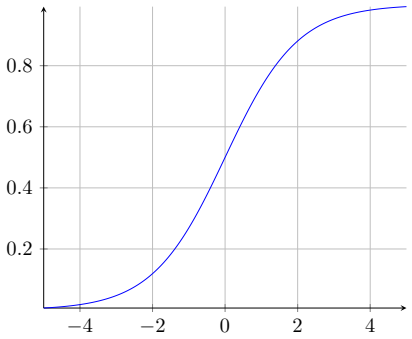
Which activation function to use in a neural network depends on the input data, output data, to which layer it is applied and what kind of learning problem needs to be solved. For the input and hidden layers it is common to use one of the activation functions that have a continuous output, e.g., *tanh*, sigmoid, softplus or LReLu. The *tanh* and LReLu functions have a negative output for a negative input which is more suitable when the input data contains negative values. The softmax activation function is used for classification models because it outputs a probability score for each class.

2.5.2 Multi-Layer Perceptron (MLP)

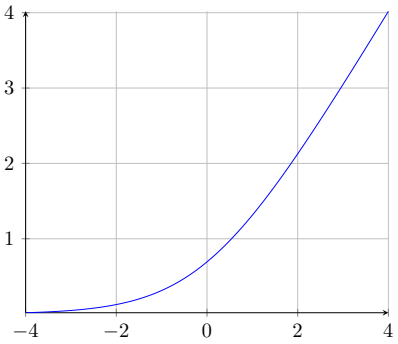
A Multi-Layer Perceptron (MLP) is one of the most basic and popular types of neural network. It consists of at least three layers: an input layer, a hidden layer and an output layer (Figure 2.4). The number of hidden layers can vary, and each layer contains a configurable amount of neurons, also called perceptrons, and the perceptrons are



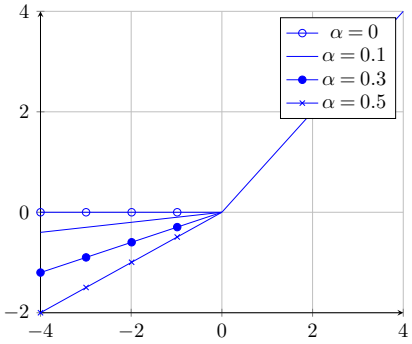
(a) Tanh



(b) Sigmoid



(c) Softplus



(d) LReLu for several values of α

Figure 2.3: Plots of commonly used activation functions

densely interconnected. Specifically, a certain perceptron from a layer is connected to all perceptrons in the next layer.

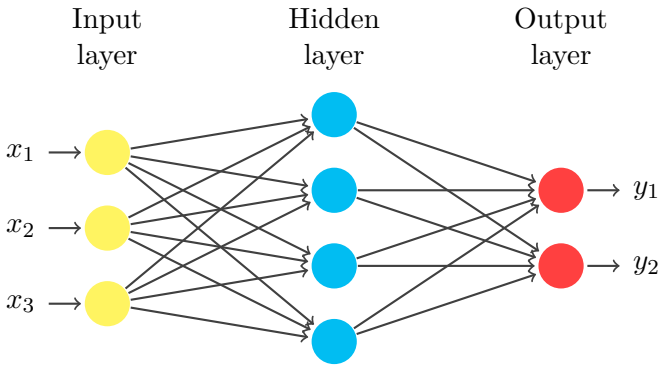


Figure 2.4: An MLP neural network with one hidden layer

The input layer is only a representation of how the input values are connected to the network, so no activation function is needed in this layer. In the hidden and output layers, each neuron j sums up the incoming signals multiplied with their respective connection weights w_{ij}. Then the activation function f is applied on the weighted summation to

calculate the output y_j of the perceptron. Frequently used activation functions are a simple threshold, sigmoid, softmax or hyperbolic tangent function. The output of each neuron is described by the following equation:

$$y_j = f\left(\sum w_{ji} x_i\right) \tag{2.6}$$

The weights of all perceptron interconnections are altered based on the error of the output signal by comparing the generated output of the network with the expected output. Again, there are several error functions, also called loss functions available, and a widely used function is the least-squared-error loss function [13]. All weights are altered to minimise the calculated error using a process called back-propagation. After an output is predicted for a certain set of input features, the delta is calculated between the ideal weights for the current output/input combination and the current weight, and a ratio of the delta is subtracted from each of the current weights, this ratio is called the learning rate. The total delta is not subtracted, to get the error to zero for this sample, to prevent over-fitting and non-converging networks. The weights are updated for each sample and by adjusting the weights only a little bit for each input/output combination, the network learns to predict based on a generalisation of all samples.

2.5.3 Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is a neural network where the internal state is also used as input for the next iteration, so that the network architecture contains a cycle [14, 15]. In this way, it is possible for information to persist in the network over time because the network now has memory, and it becomes possible to find time-related artefacts in the input data. In Figure 2.5, we can see a simple schematic of an RNN. The left part shows the folded version of the architecture containing a loop: the output of the hidden layer A is also used as input. If we unroll the loop, we get a repetitive neural network where the output of a certain time step is passed on to the next time step. The neurons itself can be a normal perceptron as used in MLPs, but specific neuron types are developed as we see in the next subsection.

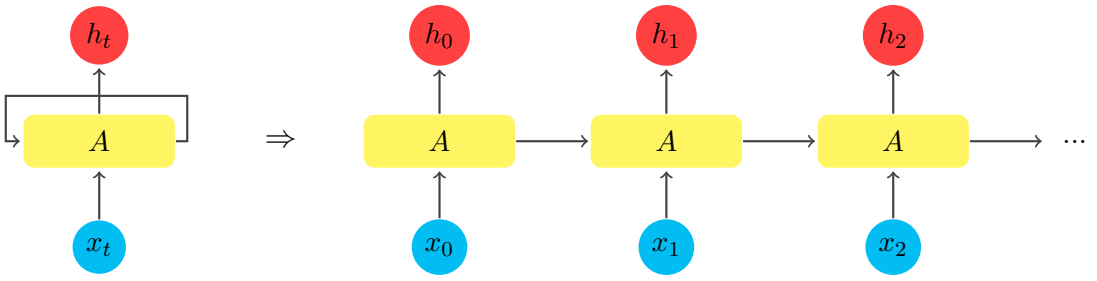


Figure 2.5: A recurrent neural network folded and unfolded

For an RNN, a special back-propagation algorithm is used named Backpropagation Through Time (BPTT) where the back-propagation algorithm is applied up to several time steps back [16].

2.5.4 Long Short-Term Memory (LSTM)

The problem with RNNs using BPTT is that, although it is in theory possible, it turns out to be hard to learn long-term dependencies [17]. When the error propagates backwards in time, weights tend to blow up or vanish, resulting in oscillating weights or very long training times. Long Short-Term Memory (LSTM) networks address this issue by using specific type of neurons, called memory cells, and are explicitly designed to avoid the long-term dependency problem. LSTMs were first introduced by [18] after which others improved upon this work, but the general idea is still the same.

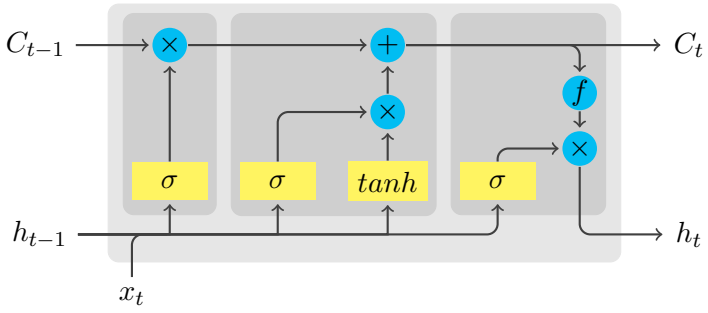


Figure 2.6: LSTM memory cell overview with gating units

Contrary to MLPs which use a single activation function on the weighted summation of the input, LSTMs use several so-called gating units to be able to forget previous learned data, remember new data and determine the output. Both the internal cell state and the output of the previous time step are fed back into the LSTM cell (C_{t-1} and h_{t-1} respectively), see Figure 2.6. Before the data is used in the gates, the previous output h_{t-1} and the new input x_t are concatenated into a new vector. Within an LSTM memory cell multiple layers can be identified:

- The first layer is a sigmoid layer, called the forget gate layer. The output of a sigmoid function is always between 0 and 1, so the output of the layer determines which values to forget, and is multiplied by the previous cell state C_{t-1}.
- The next layer of a memory cell adds new information to the cell state; for this, a *tanh* layer creates a vector of new candidates based on the input data and previous output. The input gate layer, again a sigmoid layer, decides which values to update and these values are multiplied which each other and added to the cell state.
- The last step is deciding the output of the memory cell, which is actually a filtered version of the cell state. First, another sigmoid layer decides which values to output, which is then multiplied with the output of the activation function f of the cell state.

2.5.5 Gated Recurrent Unit (GRU)

Another type of a recurrent neural network uses Gated Recurrent Units (GRUs) as neurons [19], to which we refer as a GRU network. Like with the LSTM memory cell,

the GRU has gating units that regulate the information flow within the neuron, but without the separate layers as defined for LSTM networks. A GRU only contains two gates, named the update gate and the reset gate. There is no output gate so the GRU does not control how much of its internal state is exposed to other neurons in the network and exposes its whole internal state. Another difference with an LSTM unit is that a GRU rather controls the amount of information flowing in from the previous time step than the amount of new information added. Figure 2.7 show a schematic overview of a GRU.

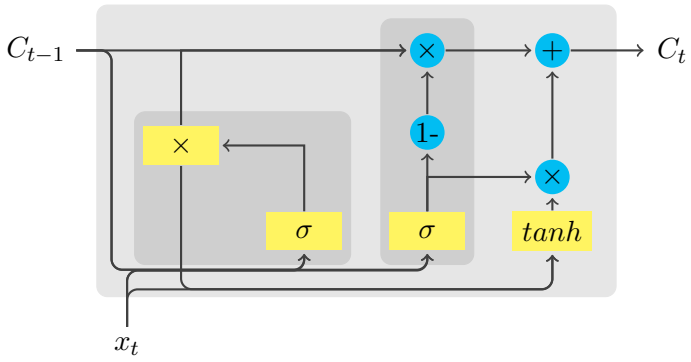


Figure 2.7: GRU overview with gating units

2.5.6 Weibull Time-To-Event Recurrent Neural Network (WTTE-RNN)

The WTTE-RNN [20] is a special type of RNN designed specifically to predict a Time-To-Event (TTE). The network uses the Weibull distribution [21], which is a continuous unimodal probability distribution with two parameters λ and k. Unimodal means that it has at most one peak. See the following equation for the distribution function:

$$f_X(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{k}\right)^{k-1} \exp\left(-(x/\lambda)^k\right) & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.7}$$

where λ and k are the two parameters that determine the distribution output.

The Weibull distribution is used in a wide range of applications, e.g., in survival analysis, where it depicts the probability of the time until failure. The distribution can also adopt the form of other distributions, e.g., the exponential-distribution (with $k = 1$) and the Rayleigh-distribution (with $k = 2$), making it a very expressive distribution in terms of possible forms with only two parameters.

The WTTE-RNN works with any type of RNN as e.g., a GRU network or LSTM network. The main properties are the combination of the activation functions, loss function and the output. The network is trained to output two values, which are the two parameters of the Weibull distribution: λ and k. These depict the probability distribution until the next event. The activation functions are custom for both output

values, the exponential and soft-plus function are used for λ and k, respectively. The loss-function to be used is the log-likelihood function.

3 Related work

In this chapter we will give an overview of previous research on seizure-signal analysis. This includes both seizure detection algorithms, see section 3.1, and seizure-prediction algorithms, see section 3.2.

For detection, we will only list research related to absence seizures, because absence seizures have different characteristics than other types of seizures and because this thesis focuses on absence seizures. For prediction, we will include multiple types of seizures because research on absence seizures has been limited and because the event leading to a seizure can be the same for multiple seizure types.

3.1 Seizure detection

As we have seen in section 2.1, absence seizures have very clear characteristics and are easily visually detectable by looking at Electroencephalography (EEG) or Electrocorticography (ECoG) recordings. This raises the expectation that it should be possible to detect seizures also automatically. In this section, we give an overview of multiple algorithms to detect absence seizures.

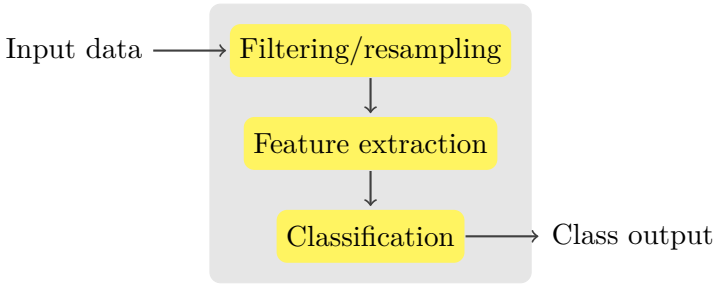


Figure 3.1: Schematic overview of a seizure detection algorithm

A seizure is a binary event: either one is ongoing in a certain time window or not. Therefore, a binary classifier can be used to identify seizures based on certain features. Multiple features and classifier algorithms have been proposed in previous research, and in general a seizure-detection system follows the schematic of Figure 3.1. First, the input data is pre-processed and features are extracted. These features are then passed onto the classifier, which decides on whether a seizure is currently ongoing or not. In this thesis we mainly focus on absence seizures, as they have specific characteristics compared to all other seizure types. Table 3.1 shows an overview of the discussed studies on seizure detection.

3.1.1 Detection using Approximate Entropy (ApEn)

One of the popular ways to detect an absence seizure is by using the *Approximate Entropy (ApEn)* of a signal. ApEn is a good measure of the complexity of a signal [7], and can also be applied to biomedical signals [8]. Complexity is an important measure of non-linear dynamic systems, and the approximate entropy is a parameter that is proportional to the complexity of a temporal signal. Compared to other parameters, e.g., sample entropy, it has several advantages [22]:

- It requires fewer data-points to calculate a sample (starting at around 100);
- It is robust against noise;
- It can be used for both deterministic-chaotic and stochastic processes.

See also subsection 2.2.2 for an explanation on how the ApEn is calculated.

Ocak [23] combines the approximate entropy with features based on the Discrete-Wavelet Transform (DWT). First, the EEG signal is analysed using DWT up to three levels and, for each of the approximation and detail coefficients, the ApEn values are calculated. The embedding dimension (m), vector comparison distance (r) and time delay (τ) were set to 2, 0.15 times the standard deviation of the data and 1, respectively. Seizure detection was done by applying a threshold to the ApEn values, where ApEn values less than the threshold were classified as epileptic. They show that the best results were obtained when using the first level detail coefficients of the wavelet transform (43.4-86.8 Hz) which yields an accuracy up to 96%.

Liang et al. [24] combine ApEn with two different features and compare the result between both combinations. The selected features are the EEG power spectrum and an Autoregressive Model (AR model) of order 20. The EEG power spectrum is calculated using a Fast Fourier Transform (FFT), and it is shown that during a seizure a larger power is measured in the 7-9 Hz band, which is used as the detection feature. The AR model is a parametric model for describing stationary time series and contains the weighted sum of previous values and white noise. Classification is done using both linear and non-linear classifiers: Linear Least Squares (LLS), Linear Discriminant Analysis (LDA), Back-propagation Neural Network (BPNN) and Support Vector Machines (SVMs). The combination that on average yields the best result uses the ApEn feature combined with power-bands features and a level Principle Component Analysis (PCA) and has a sensitivity, specificity and accuracy of 97.03%, 97.83% and 97.50%, respectively.

Another combination of features with ApEn is found by Sakkalis et al. [25], which combine ApEn with the Order Index (OI) and Multiscale Variance Index (MVI). The Order Index is a quantification of the degree of order of a non-stationary time series e.g. an EEG signal. The MVI describes the variance of a signal around its arithmetic mean. It is shown that during a seizure the OI is significantly lower and the MVI increases. Classification is done using a threshold. ApEn yields the highest sensitivity, while OI and MVI yield the highest results for specificity. The overall highest accuracy is obtained using ApEn with an accuracy of 90.1%.

3.1.2 Detection using Wavelet Transform (WT)

Another popular method for seizure detection is the use of a *Wavelet Transform (WT)* to extract features from an EEG or ECoG signal. With the WT, one can do a frequency analysis, but with a varying window length. This results in signals with varying frequency and time resolutions, where the frequency resolution is inversely proportional to the time resolution. In contrast to FFTs, it is also possible to choose the basis-function, where an FFT always uses the sinusoid function. A commonly used, so-called mother wavelet is the Daubechies function, which is scaled and shifted in time in order to decompose the signal into subbands of different frequencies. The discrete WT is implemented using a high- and low-pass filter, resulting in a low-frequency approximation band and a high-frequency detail band. For each consecutive level, the approximation band is further decomposed into another approximation and detail band [26], see also subsection 2.2.1.

Subasi published two approaches on using DWTs for detecting seizures in EEG signals [27, 28]. Both approaches use the DWT, but features and classification differ.

In the first approach a Daubechies 4 (DB4) wavelet is used to filter the EEG signal. The number of levels was chosen to be 5, because EEG signals do not have any useful frequency components above 30 Hz. Classification is done using Artificial Neural Network (ANN), where two solutions are compared: BPNN, here called feedforward error backpropagation ANN (FEBANN), and a Dynamic Wavelet Neural Network (DWNN). The input features for the neural networks were the wavelet coefficients of the four subbands (α, β, δ, θ). The best results were obtained using the DWNN with an accuracy, specificity and sensitivity of around 93% [27].

The second approach also used a 5-level DWT, but instead of using the coefficients directly, it uses statistical features based on the wavelet coefficients. The following features were extracted:

1. Mean of the absolute values of the coefficients in each sub-band.
2. Average power of the wavelet coefficients in each sub-band.
3. Standard deviation of the coefficients in each sub-band.
4. Ratio of the absolute mean values of adjacent sub-bands.

Classification is done using a Multi-Layer Perceptron (MLP) and an Adaptive Neuro-fuzzy Inference System (ANFIS) after which the results are compared. An ANFIS uses the neural-network training-process to generate a set of fuzzy if-then rules, that approximate a desired data set. The ANFIS approach makes for an accuracy of about 94% [28].

Petersen et al. [26] also utilised the WT but use the log-sum energy of 6 wavelet subbands as features. Classification is performed using an SVM. To prevent false positives, a temporal filter is used so that at least 3 consecutive epochs (2 seconds) must be classified as ictal before a seizure is detected.

3.1.3 Other features

Other approaches can also be found in literature. Zeng et al. [29] use the Normalised Mean Squared Error (NMSE) and Structural Similarity) (SSIM) at a certain compression rate of the EEG signal and classify these features with several classifiers: Decision Tree (DT), K-nearest Neighbor (KNN), discriminant analysis (DA) and SVMs. Sensitivity and specificity are both around 72%.

Alkan et al. [30] calculate power spectra of three different methods, multiple signal classification (MUSIC), AR model and periodogram methods, and use these as the inputs to a classifier. The classifiers used are a statistical method based on Logistic Regression (LR) and an ANN. The best results were obtained using a MLP in combination with the MUSIC feature with an accuracy of 92%, specificity of 93.6% and sensitivity of 90%.

Diambra and Malta [31] use a non-linear custom method based on information theory to detect ictal Spike-and-Wave Discharge (SWD) activity. A part of the recording of each patient is used to determine the characteristics of the EEG signal for that particular patient, which is used to create a non-linear model of the signal. The error of the predicted samples of the model is an indication for ictal spikes, as they have different characteristics than the inter-ictal EEG signal, which does normally not contain any SWDs. No accuracy is reported.

3.2 Seizure prediction

The prediction of seizures is a totally different working field compared to detection as the inducement of a seizure is not clearly visible in the brain signals. More rigid methods and algorithms are needed to be able to make a proper forecast on if and when a seizure will occur. A lot of research has been done on this field, but in particular for absence seizures, no solid solution has been found. There are however studies that found signs of precursors in the brain for seizures, e.g., [32] found a positive correlation of the seizure duration and intensity with β-power (20-40 Hz) and a negative correlation with θ-power (4-7 Hz), see also subsection 3.2.2.

In this section, we will look at previous research done in the field of seizure prediction. Because the amount of research targeted at absence seizures is limited, we will also refer to research done in general and targeted at tonic-clonic seizures. It is until now not totally clear what leads to a seizure, therefore the same algorithms could possibly work for both absence and tonic-clonic seizures.

3.2.1 Prediction studies

The first works date back to the 1970s when different groups started using linear approaches on EEG data, showing promising results. They however used short and manually selected EEG recordings. Later research found substantially poorer results than promised by the first approaches, and some previous research could not be reproduced, while using larger data sets [33]. In the years that followed, more methods have been explored and used for predicting seizures. See Table 3.2 for an overview.

Year	Author(s)	Features	Classification	Recordings			Performance[a, b]	
				Type	# Patients	Seizures	TPR	TNR/FDR
1999	Diambra and Malta	Prediction error of non-linear model	-	EEG	3[c]		-	
2005	Alkan et al.	MUSIC, AR model, periodogram	LR, ANN	EEG	5	20	90%	93.6%
2005	Subasi	DWT coefficients	BPNN, DWNN	EEG	5	20	92.8%	93.1%
2007	Subasi	Statistic DWT features	MLP, ANFIS	EEG	5	20	94.3%	93.7%
2009	Ocak	ApEn, DWT coefficients	Threshold	ECoG	5		96%	93.7%
2010	Liang et al.	ApEn with signal power or AR model	LLS, LDA, BPNN, SVM	EEG	3[d]	1758	97.8%	98.4%
2011	Petersen et al.	WT log-sum energy	SVM with temporal filter	EEG	19	111	99.1%	0.5/h
2013	Sakkalis et al.	ApEn, OI, MVI	Threshold	EEG	8	75	97.33%	83.91%
2016	Zeng et al.	NMSE or SSIM of compressability	DT, KNN, DA, SVM	EEG	9	200s	Accuracy: 76.7%	

[a] Maximum reported performance of a certain combination of features and classifier
[b] TPR: True Positive Rate, also sensitivity, TNR: True Negative Rate, also specificity
[c] Of which 2 have focal epilepsy
[d] Recordings done on rats instead of humans

Table 3.1: Overview of studies on seizure detection

Netoff et al. [34] report a seizure-prediction project where they utilised an SVM and used the Freiburg EEG dataset [35]. Using visual inspection of the signal, recordings with unwanted artefacts were removed. Then power features from 6 different recording electrodes in 9 different spectral bands were extracted, resulting in 54 features in total. Classification was done using an SVM and yielded a True Positive Rate (TPR) of nearly 80% and a False Positive Rate (FPR) of 0/hour.

Chisci et al. [36] describe another prediction method using AR model coefficients as features and also used the Freiburg EEG dataset [35]. They utilised the coefficients of an AR model which is trained specifically for each of the 9 selected patients. Classification is done using a SVM regularised by means of a Kalman Filter (KF). The system yields a TPR of 100% and an average FPR of 0.41 seizures/hour.

Park et al. [37] used a combination of the aforementioned approaches while also using the same dataset. After manually removing artefacts in the data, bipolar and/or time-differential methods have been used to remove or reduce the effect of other types of artefacts in the ECoG data. Spectral power features were then extracted from 9 bands using a 20s-long half-overlapping sliding window. Classification was done using a cost-sensitive SVM and finished off with a post-processing step utilising a Kalman Filter to remove isolated false positives. The best results were obtained using the bipolar method and yielded a TPR of 97.5% with a FPR of 0.27/hour.

Gadhoumi et al. [38] used recordings from 17 patients with in total 1656 hours of ECoG data. Features were extracted by calculating the wavelet energy and entropy in different frequency bands, using a 2-second non-overlapping sliding window. Different states were identified using discriminant analysis and in-sample cross validation. With a seizure occurrence period above 30 min, the method performed above chance with TPRs higher than 85% and FPRs below 0.1/h.

Howbert et al. [39] present another approach using the spectral power in multiple ECoG frequency bands. 6 bands were used, and the power was summed in a 1-minute non-overlapping sliding window for each of the 16 electrodes, resulting in a feature vector of 96 features for each 1-minute block. Logistic Regression (LR) classifiers were trained to classify the blocks as being pre-ictal or inter-ictal. The pre-ictal time was chosen to be 90 minutes, and also the seizure itself was labeled as pre-ictal. On average a TPR of 70% and a FPR of 0.08/h is obtained.

Alexandre Teixeira et al. [40] did an extensive research on long-term seizure prediction with data from 278 patients. The EEG data was recorded using 22-37 electrodes, but recordings from only six electrodes were selected for this research. Windows of 5 seconds were used, on which 22 features were computed. These features included the AR model predictive error, decorrelation time, energy, complexity, relative power, spectral frequency and power, statistical features and DWT coefficients. Classification was tested using two different classifiers: ANNs and SVMs. For 32% of the patient seizures could be predicted with a TPR larger than 50% and a FPR of less than 0.15/hour.

Moghim and Corne [41] presented another approach, named ASPPR, evaluated on the Freiburg EEG dataset [35]. They also first removed unwanted artifacts, after which a total of 204 features were extracted, 34 distinct features for each of the 6 ECoG channels. Features were based on the signal energy, DWT coefficients and non-linear dynamics. For each patient the large feature-set was reduced to the best working 14 features, selected

using the ReliefF algorithm. A multi-class SVM was then used to classify the several states of the ECoG signal. The highest result was obtained for a prediction time of 20 to 25 minutes with a TPR of 90.15% and a True Negative Rate (TNR) of 99.44%.

Alvarado-Rojas et al. [42] reported an approach where they used the coupling between low-frequency phase and high-frequency amplitude, which can be used to distinguish between pre-ictal and non-ictal states. Recordings were done on 52 patients with partial epilepsy, and for each patient about ten days of data was recorded. On average, there were 11 seizures per patient recorded. Short-time phase fluctuations were smoothed using a first-order Kalman Filter (KF). Classification is then performed using a variable threshold resulting in very varying results per patient. Results vary from a sensitivity of 0% up to 100% and FPRs of 0.02/h to 6.02/h. On average, the presented work yields a TPR of 46.5% and a FPR of 0.94/hour. The coupling between low-frequency phase and high-frequency amplitude is shown to be existing for a significant number of the 52 patients (13.2%).

3.2.2 Predictability studies

Other works only discuss the predictability of certain metrics without evaluating the results, or without performing the actual seizure prediction. We will discuss a few of these publications as they can provide interesting features for our work.

Ouyang et al. [43] tried to understand the transition of brain activity towards an absence seizure and applied the determinism measure (DET) for this task. They found that the DET value distribution was significantly different for pre-ictal, ictal and inter-ictal periods, leading to the tentative conclusion that a higher degree of determinism is present during pre-ictal time windows.

Sorokin et al. [32] presented an effect that precedes a seizure. They found a positive correlation of the seizure duration and intensity with β-power (20-40 Hz) and a negative correlation with θ-power (4-7 Hz). Also, they report a decrease in thalamocortical neuron spiking frequency during the pre-ictal period. In addition, they discovered that the β-power before a seizure inducing optical stimulation negatively correlated with the SWDs onset.

Lüttjohann et al. [44] described their research on finding alpha, delta and theta precursors before SWDs onset. Discovered was that a SWD is preceded by precursors consisting of several frequency components ranging from 2 to 12 Hz. In general, alpha and theta activity was found in the frontal cortex and thalamus simultaneously, but delta events appeared first in the cortex and then in the thalamus after a delay. In around 90% of the cases precursors were present in the cortex, and in around 82% of the cases they were present in the thalamus. The average precursor duration was 0.5s.

3.2.3 Prospective versus retrospective

The first prospective studies date from 2003 (Iasemidis et al. [45], D'Alessandro et al. [46]) and are the first which tried to predict a seizure in the future only based on past data. All other research done until that point looked retrospectively at brain recordings, which means that they looked for features in the data before a seizure when the seizure already occurred. Mapping events in the data by looking only at recordings just before

a seizure does not mean that these events only happen when a seizure is following. This could easily lead to false-positive prediction because events that happen before a seizure, but also during normal brain activity, can be wrongly seen as seizure precursor. This makes the prediction problem a lot harder than when retrospectively looking at the data.

This work will focus on prospective prediction of absence seizures, by only looking at past data at a certain point in time.

Year	Author(s)	Features	Classification	Recordings			Performance[a, b]		
				Type	Patients	Data	TPR	FPR	TNR
2009	Netoff et al.	Spectral power of 9 bands	SVM	ECoG	9	45 seizures, 219h interictal data	77.8%	0/h	
2010	Chisci et al.	AR model coefficients	SVM with KF	ECoG	9	~ 40 seizures	100%	0.41/h	
2011	Park et al.	Moving-window power of 9 bands	SVM with KF	ECoG	18	>54	97.5%	0.27/h	
2013	Gadhoumi et al.	Wavelet energy and entropy	DA, in-sample cross validation	ECoG	17	175 seizures	>85%	0.1/h	
2014	Howbert et al.	Spectral power of 6 bands	LR	ECoG	3[c]	125 seizures	70%	0.08/h	
2014	Alexandre Teixeira et al.	22 univariate features	ANN, Multi-class SVM	EEG	278	1519 seizures	>50%	<0.15/h	
2014	Moghim and Corne	$34 \cdot 6 = 204$ univariate features (reduced to 14)	ReliefF, Multi-class SVM	ECoG	21	-	90.15%		99.44%
2015	Alvarado-Rojas et al.	Coupling of phase and amplitude	Variable threshold	ECoG	53	558 seizures, 531 days data	46.5%	0.94/h	

[a] Maximum reported performance of a certain combination of features and classifier
[b] TPR: True Positive Rate, also sensitivity. TNR: True Negative Rate, also specificity. FPR: False Positive Rate.
[c] Recordings done on dogs instead of humans
[d] For 32% of the patients

Table 3.2: Overview of studies on seizure prediction

4 Design and Implementation

In this chapter, we will describe the design and implementation choices for this thesis work. The goal is to study the feasibility of a system able to predict whether a seizure is about to occur, within a certain time window, which is referred to by the term *pre-ictal time*. As will be further explained in chapter 5, the results turned out to be negative in most cases. Some approaches only yielded detection results, because an Artificial Neural Network (ANN) can distinguish between ictal and non-ictal data easily, but prediction of a upcoming event based on Electrocorticography (ECoG) measurements is a lot harder. Therefore, we can say that a prediction time of zero seconds or less is actually a seizure detection.

For the experiments done in this work, ECoG measurements of mutant mice are used, as further described in section 5.1. To predict the occurrences of absence seizures we will apply several types of ANN algorithms. The input data has to be pre-processed to be in a for the ANN usable format, which is done by normalising the data and by extracting useful features using dedicated algorithms. The features being used are the power of the θ, α and β Discrete-Wavelet Transform (DWT) bands, the Approximate Entropy (ApEn) and the signal variance, see also section 4.2. The output of the network also needs to be post-processed and has to be analysed for determining the performance of the system. The experiments can be distinguished based on the type of input- and output-data and network type, and we follow the design path as shown in Figure 4.1. The experiments are performed independent of each other and will be evaluated in the next chapter.

First, the experiments are divided based on the two output types. The output type for the first set of experiments are classes, encoded as a one-hot class number. With a classification ANN it is possible to classify input data into multiple classes. This matches our goal as we want to classify inter-ictal, pre-ictal and ictal periods in the input data. The second set of experiments has the Weibull distribution parameters as output, which is used for Time-To-Event (TTE) prediction. With a Weibull Time-To-Event Recurrent Neural Network (WTTE-RNN) a time is predicted until the next seizure will happen. This type of network is selected because it offers a possibility of using regression for time prediction. The training parameters and evaluation differs for both output types and therefore experiments are grouped accordingly.

For the classification experiments we will then apply three different types of ANNs, namely Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), and for Weibull Time-To-Event (WTTE) prediction we will only use the recurrent versions LSTM and GRU. For each network type two input data-sets will be used, the first being the normalised recordings and the second being a set of extracted features, further explained in the next sections. The items shown in blue are already covered extensively by other research. The yellow boxes are also explored by

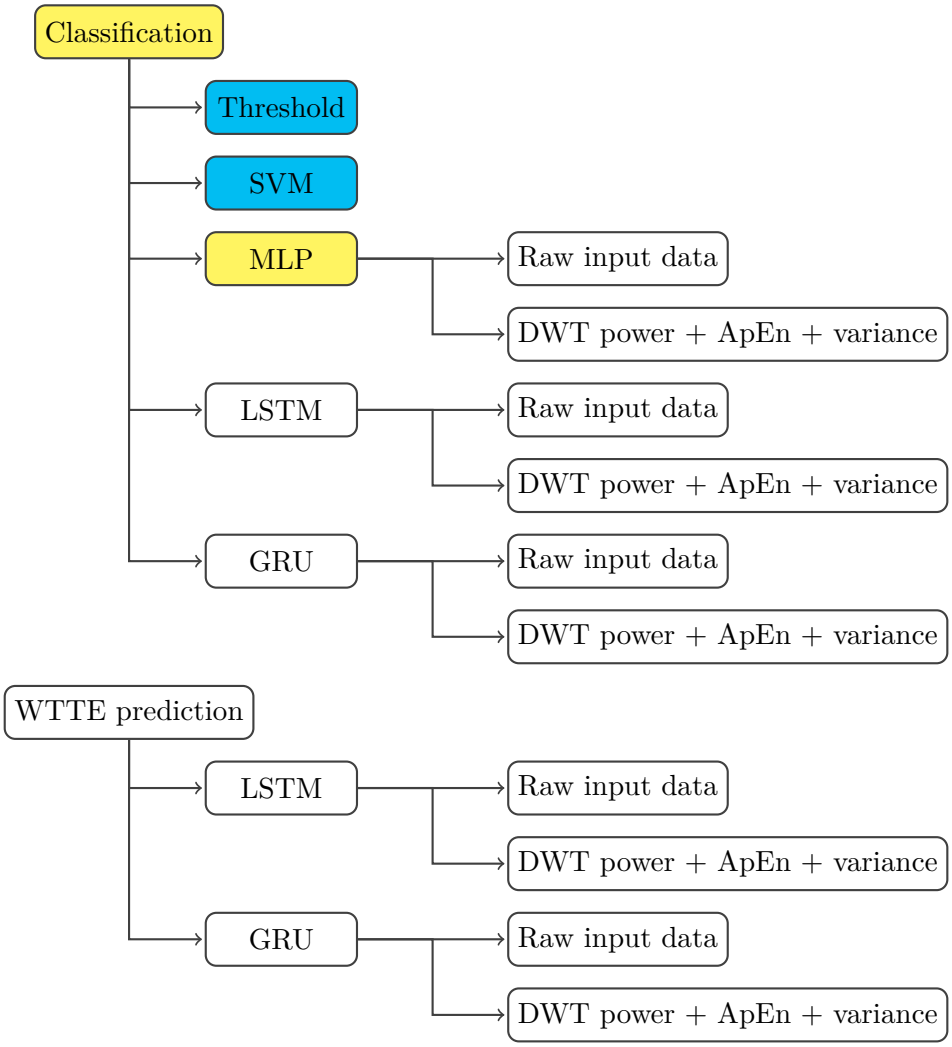


Figure 4.1: Overview of the experiments performed in this thesis, specifying the output type, network type and input data. Blue boxes are covered by other research and will not be repeated, yellow items will be applied again although being explored by other research. White boxes are new methods in the field of seizure prediction.

other research, but we will look further into these because they are suitable for our research and can give new insights in combination with our data.

First, we will discuss the choices for machine-learning algorithms in section 4.1 and then the data pre-processing for the machine-learning algorithms of choice will be discussed in section 4.2. In section 4.3 we will describe the parameter space for the ANNs and the last section (4.4) will report on post-processing of the output data.

4.1 Machine-learning algorithms

In this section, we will look at several machine-learning algorithms and motivate why they are being used in this work. There are two main features which need to be supported by the chosen machine-learning algorithms, the possibility to learn complex patterns and the possibility to learn time dependencies.

The first selection criterion, the ability to learn complex patterns, results in the need for support for non-linearity. With a non-linear algorithm, more complex relations between input and output can be learned, while a linear approach is limited to only linear relations. The property of non-linear functions is that the sensitivity to changes in the input is not the same over the whole range. This is expressed by the derivative of the used function, for a linear function the derivative is constant, while for a non-linear function this is not constant.

Non-linear behaviour is especially wanted when using ANNs, because the output of a previous layer is used by the next layer. Using a linear activation function (meaning linear over the whole domain) would result in multiple layers of neurons which together learn a linear relation between input and output data which can be replaced by one layer. Several linear functions can be added together to one linear function after all. As seizures and ECoG-measurements show highly non-linear behaviour, we need to select non-linear activation functions, which also allows us to use deep learning. The possibility of deep learning, meaning a layered application of the learning algorithm, is important, because it will allow for learning more complex patterns. Related work has also shown that it is better to use a non-linear approach; Most listed prediction studies with a higher Average Detection Rate (ADR) in subsection 3.2.1 use a non-linear learning algorithm which acknowledges the choice for a non-linear approach.

Further, we also want to implement a way to learn time dependencies from the data. The precursors to a seizure can be present in the data a while before the actual onset, and build up gradually [47]. The output of the algorithm should be either multi-class or a continuous regression value.

For classification we want to be able to distinguish between three classes, identifying the inter-ictal, pre-ictal and ictal periods, which is not possible with a binary classifier. All three ictal classes have to be predicted, so we can fall back on seizure detection in case a pre-ictal period is not predicted for a particular seizure. For regression we want to be able to predict the time to the next event, which is a continuous value.

4.1.1 Support Vector Machine (SVM)

An SVM is a binary classifier of which also multi-class versions are available; see section 2.4. We have seen multiple uses and different variations of SVMs in related work. Although non-linear multi-class SVMs exist, we will not use this classifier in our work, because SVMs do not take in account time dependencies in the data while we expect this to be important for our work, and because SVMs cannot be used as a regressor, which we need for TTE predictions. However SVMs have been used a lot in related works as classifier, our works demands for more advanced classifiers which also support deep learning.

4.1.2 Multi-Layer Perceptron (MLP) neural network

Another type of learning algorithm we have seen in literature are ANNs and especially the MLP variation. See also subsection 2.5.2 for more background on this type of neural network.

An MLP is fed a list of input data, which can be a list of samples spanning a time window or a feature-set. A combination is also possible resulting in a matrix of input data, which has to be flattened before the neural network can use it, meaning that the matrix is reshaped into a one-dimensional list. By spanning a time window it is possible to make the algorithm learn time dependencies in the input data and a sliding and partially overlapping window can be used to cover more of the input situations.

The activation function of the neurons in an MLP determines the output of a neuron based on the input. First, the input is multiplied with the input weight and then summed, after which the activation function is applied. The activation function of hidden layers is always non-linear because otherwise deep learning is not possible. MLPs meet the requirements to be included in this research and will therefore be evaluated.

4.1.3 Long Short-Term Memory (LSTM)

Another type of ANNs is the Recurrent Neural Network (RNN), where recurrent means that is uses also its internal state as an input for the next evaluation step in the network. This means that the internal state contains information from the input from previous timesteps. Input from the past has therefore influence on the current output, resulting in the ability to learn across time. This is particularly important for predicting seizures as we do not know when a precursor is present in the signal. A further advanced version of a general RNN is the Long Short-Term Memory (LSTM) neural network, further explained in subsection 2.5.4.

An LSTM is specifically designed to be able to learn time dependencies from input data, and does this using several gating units in each neuron dedicated to learning new information, forgetting already learned information and by keeping learned information. What information to learn and what to forget is determined during the learning process based on back-propagation as is done with other neural networks.

Because an LSTM already has a built-in way of dealing with time dependencies, we do not need to feed large time windows to the network. The input data only has to span the feature-set, meaning that we can use the features of only one point in time as input. For MLP networks, we have to use input data which also spans a time-window, so use features of multiple points in time because it has no internal way of relating inputs with each other over successive runs. The output of the system can also be a discrete class or a regression value, which is the same as for MLPs. Also, the activation functions used for MLPs, as mentioned above, can be used for LSTMs too, making for a non-linear system.

These properties, the native ability to learn time dependencies, the non-linear behaviour and the output types, make LSTMs a good candidate for evaluation in this thesis.

4.1.4 Gated Recurrent Unit (GRU)

A later successor of the LSTM neural network is a RNN which is named Gated Recurrent Unit (GRU), see also subsection 2.5.5.

A GRU network implements the way of learning and forgetting data in a slightly different way, but the general idea is the same. It is also a drop-in replacement for an LSTM meaning that we can use this type of network without changing the structure of the implementation. Therefore, we will also evaluate this learning algorithm, and compare the results to those generated by LSTMs.

4.1.5 Weibull Time-To-Event Recurrent Neural Network (WTTE-RNN)

The WTTE-RNN is a RNN designed for predicting times-to-events, see subsection 2.5.6. Because we also want to predict events, seizures in this case, this approach is considered as very suitable. The two output values of the neural network correspond to the two parameters of the Weibull distribution: λ and k. When training the network, we train λ to output the time to the next event, and k is trained to depict the probability that the event occurs. In practice, this means that k is trained to be 1 if an event is upcoming, and 0 if that is not the case. So, the probability that an event will occur is 1 if we know the TTE during training.

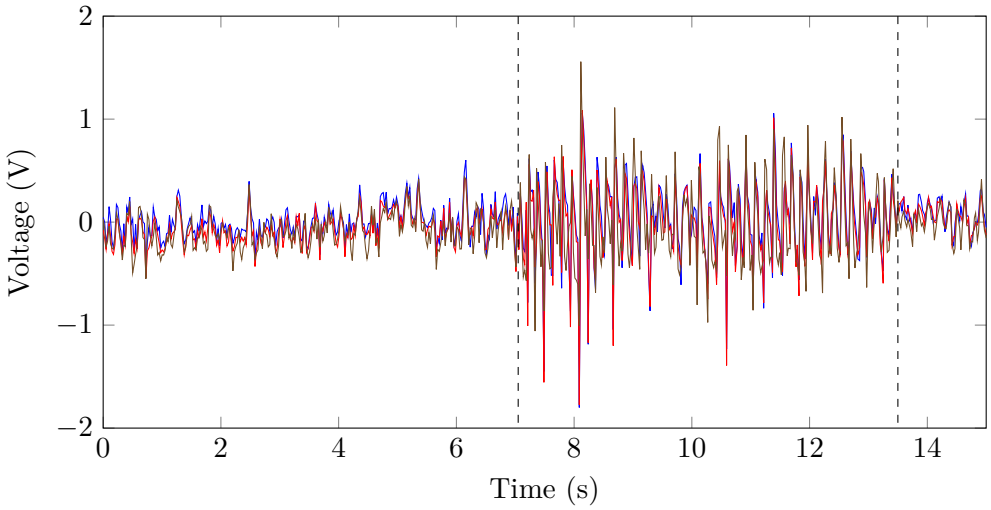
Under the hood, the WTTE-RNN can use different types of RNNs. We will use it in combination with the RNNs as listed above: the LSTM and GRU RNNs.

4.2 Input data pre-processing

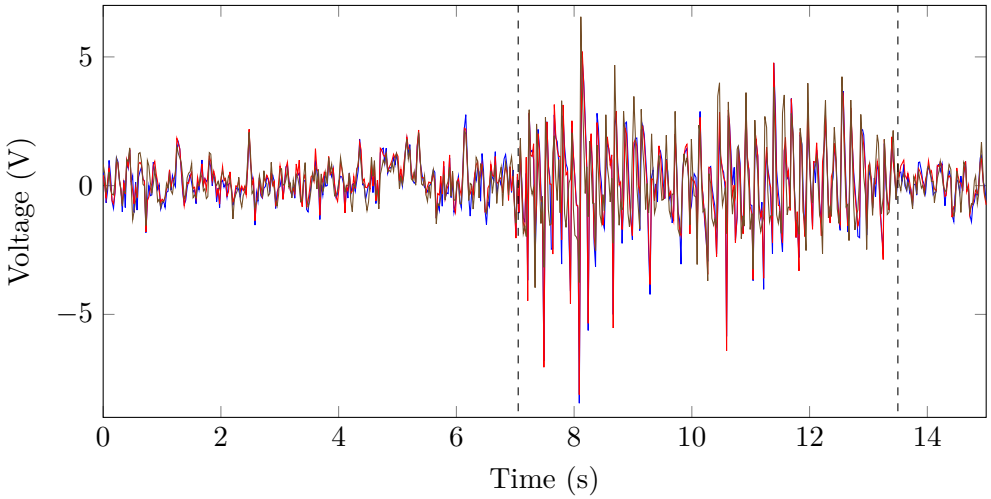
In this section, we will look at the pre-processing of the input data. The available input data was recorded for previous research using multiple electrodes on the brain of a mouse, after which channels with usable data were selected by the researchers and the data was resampled at 300 Hz [4]. See also section 5.1 for more information on the recordings.

For each pre-processing step, we will plot a graph with the result, based on a small part of the raw input data, as shown in Figure 4.2a. This to show the effect each step has in our analysis pipeline. The vertical dotted lines indicate the start and stop of an absence seizure.

As we have seen in chapter 3, there are multiple features that can be extracted using conventional signal-processing methods. Besides normalising the input data, we will use the following popular or promising features: power of the θ, α and β DWT bands, ApEn and signal variance. These features showed propitious first results in predictability analyses (see subsection 3.2.2) or have already been used in prediction studies, further explained in the following sections. They are generated from the normalised data as described in subsection 4.2.1 and therefore use the same way of balancing the output classes.



(a) Raw input data with 3 channels at 300Hz. Each line is the recording of one channel.



(b) Input data after normalisation with zero mean and unit variance

Figure 4.2: Input data before and after normalisation

4.2.1 Data normalisation

In order to use the input data for learning, we have to remove unwanted artifacts that can distort the learning process. First, we need to remove any offsets so that the data has a zero mean. Activation functions have different behaviour for negative values than for positive values. E.g., the Rectified Linear units (ReLu) activation function is most sensitive between -2 and 2, so when training a network which uses this activation function in the input layer, it would be harder to learn something when the majority of the data would be out of the sensitive range of the function. That is why we need to remove any offsets from the data which are present over the timespan of a few seconds. The offsets in the data can be the result of recording problems as e.g., a not properly grounded setup

or moving electrodes, recordings are done in a living animal after all. Removing the offset is done by subtracting a moving average with a large time-window of 10 seconds. It's not possible to subtract a fixed offset because the offset for each recording varies over time and per channel. Removing a dynamic offset will introduce some noise, but this will be less than keeping the offsets in the data as they are noise themselves.

As the next step in the normalisation process we also need to scale the input data to unit variance. The first reason for this is that activation functions are sensitive in a certain range, values out of this range are treated all the same. This is especially the case for asymptotic functions as e.g., the *sigmoid* and *tanh* activation functions, see Figure 2.3 (page 12) for a plot of these functions. The second reason is that separate recordings can have a slightly different amplification which is corrected by the scaling process. Scaling of the input data to unit variance means that a certain distribution of the input data will be in the range $[-1, 1]$ after scaling. This results in a signal where a small subset of the data points, including outliers if present, have a value out of the sensitive range of the activation function. Depending on the activation function, these values will then have less influence, or no influence at all, on the output. This is convenient as outliers are unwanted artifacts in the data. Scaling is done on a recording basis, so the characteristics of each recording are comparable. One recording is recorded during the course of an experiment and has the length of about an hour. If some recordings would have very different characteristics, they would influence the learning process negatively.

Finally, because the inter-ictal and pre-ictal periods are a lot longer than the actual seizures, classes are very unbalanced when using classification. To overcome this problem there are two solutions. One is to limit the number of input samples of inter-ictal and pre-ictal classes by removing them from the signal (under-sampling). The other solution is to give the output samples a weight inversely related to the amount of samples that are available for a certain class. This weight results in a higher learning rate for these samples, so the network learns to predict the less occurring classes as good as the more occurring classes, but based on less samples. A combination seems to be the best solution as large weight differences yield worse learning results, and removing a lot of features makes for less training data and creates an unnatural data-set [48].

This normalised data-set is used for the first group of experiments, but is also used to generate the features from as described in the following sub-sections. This is done to minimise the differences in pre-processing and therefore making the results more comparable. Also the classes are balanced in the same way for all experiments, so only the input data itself is different.

4.2.2 Discrete-Wavelet Transform (DWT) power

By using a Discrete-Wavelet Transform (DWT), we can extract certain frequency bands from the input signal. The DWT can be applied at several levels, where each level produces a low-frequency approximation band and a high-frequency details band. Each level deeper, the approximation band is used as input, and in turn split into an approximation and details band. See also subsection 2.2.1 and Figure 2.2 (page 7).

Related work has shown that absence seizure susceptibility correlates with pre-ictal β oscillations [32] and that the δ, θ and α bands can also be used to predict (and detect)

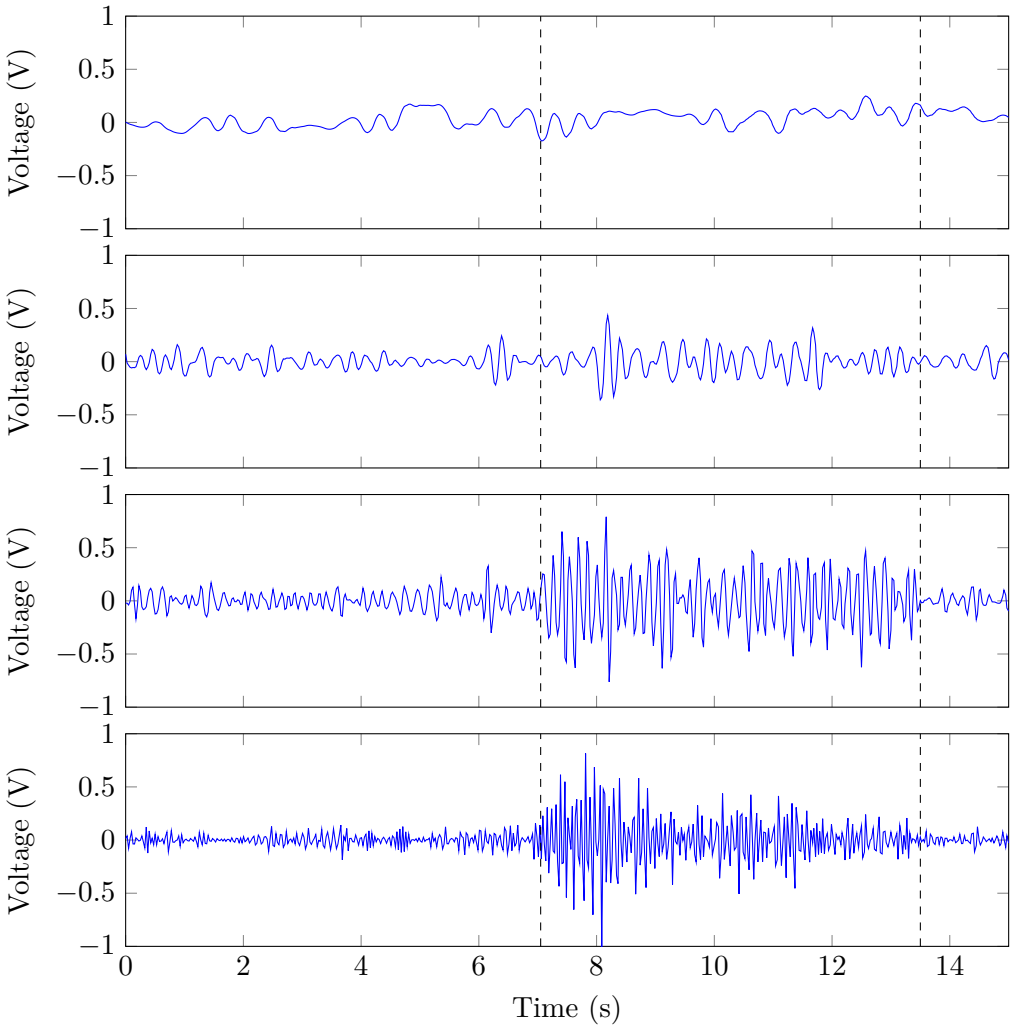


Figure 4.3: DWT data containing δ, θ, α and β bands

seizures [40, 44, 37]. We therefore have to choose a DWT level so that the resulting frequency bands overlap with the ranges of the α and β, δ and θ bands. Our input data has a frequency of about 300Hz, so after applying a DWT of 6 levels, we get the following frequency bands:

- 0 - 4.69 Hz
- 4.69 - 9.38 Hz
- 9.38 - 18.75 Hz
- 18.75 - 37.5 Hz
- 37.5 - 75 Hz
- 75 - 150 Hz

- 150 - 300 Hz

The δ, θ, α and β bands are defined as 0 - 4Hz, 4.5 - 8Hz, 8.5-12Hz and 12-36Hz respectively. The resulting DWT bands 1 to 4 approximately overlap with the needed bands and will therefore be used. The wavelet used is the Daubechies 4 (DB4) wavelet, which is used by other publications and turns out to be a well suited wavelet. The DB4 wavelet has a similar form as the Spike-and-Wave Discharges (SWDs) that occur in the ECoG signal. Also, it preserves the energy in the signal, making it suitable for our study, as we also want to use the energy in the selected frequency bands as training features for our model. Comparing different wavelets, the Complex Morlet wavelet would also be a good candidate and is generally a good choice for time-frequency analysis, but we are more focused on filtering the signal than on time-frequency analysis. See Figure 4.3 for a plot of the δ, θ, α and β bands. For simplicity, only one channel is shown for each band.

We use the power of each frequency band because it is a measure of the brain activity in that particular band. The power is calculated by squaring the values, see Figure 4.4 for a plot of the power of the data shown in Figure 4.3.

4.2.3 Approximate Entropy (ApEn)

Approximate Entropy (ApEn) is a good measure of complexity or regularity of a signal, and an indication of the unpredictability of fluctuations in the data [7]. It has been developed especially for determination of the regularity of biologic signals, or other natural appearing signals, containing white noise. Related work has shown that it is a usable metric for seizure detection, and maybe also prediction; see subsection 3.1.1.

The calculation of ApEn requires three parameters: the embedding dimension (m), noise filter level (r) and data length (N), see subsection 2.2.2 for a description of the parameters. There is no specific guideline for the determination of the optimal parameter values, but the procedure described in [7] is a good rule of thumb and therefore we use the parameters as described there. A further study on the parameters for brain signals is given in [6] on which we based our parameter choices. They used $N = 512$ for the data-length parameter as a good trade-off between stability of the output signal and computational time. Further, an embedding dimension (m) of 2 was chosen after examining the data with the false-nearest-neighbours (FNN) [49] algorithm. For the noise level r, we used a fixed value of 0.15 based on empirical tryouts. See Figure 4.5a for the ApEn for each of the channels of the normalised input data. The used input data chunk is shown in Figure 4.2a.

4.2.4 Variance

The variance metric describes the statistical variance in a signal, and shows how a set of values is distributed around their mean value. To calculate the variance of a certain set of data, first the mean value is determined. Then, for each data-point the squared difference is computed, and the average of these values is the variance. The parameters for this algorithm are the window length N and the sliding window step s.

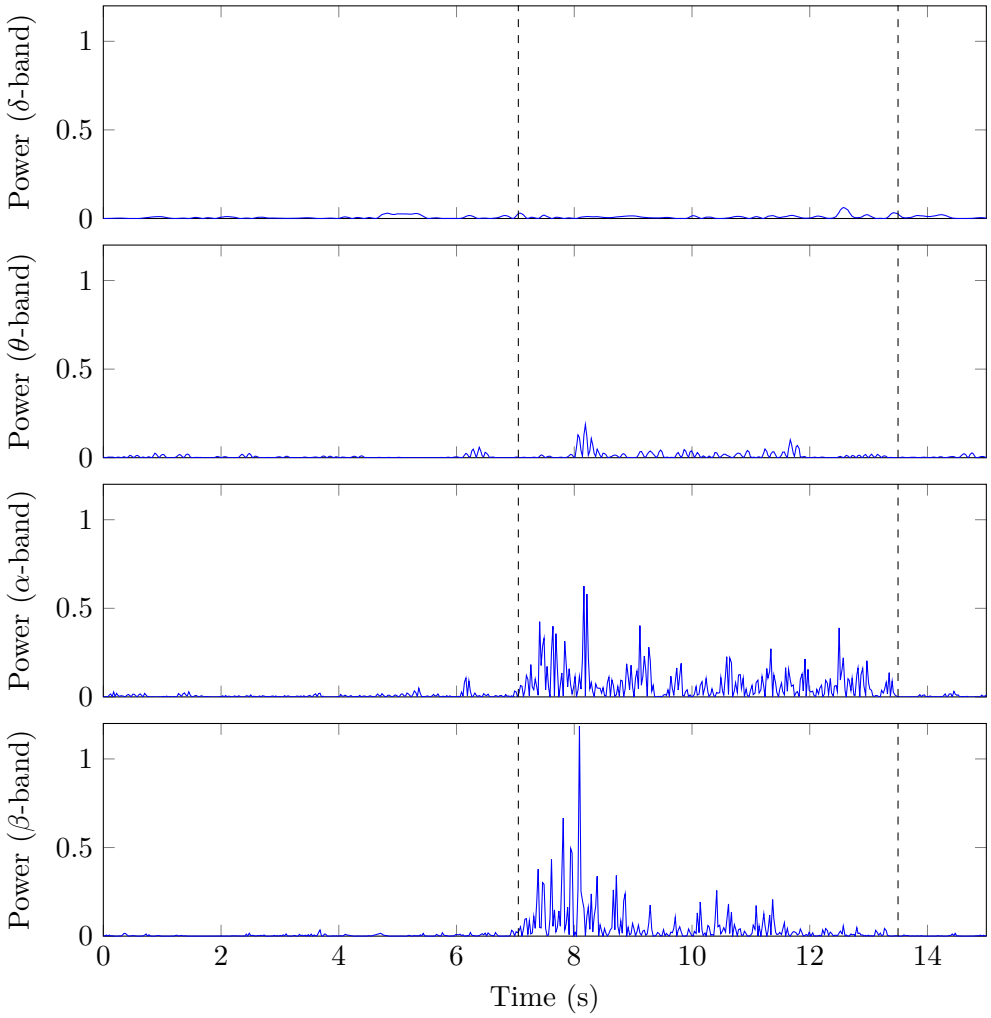


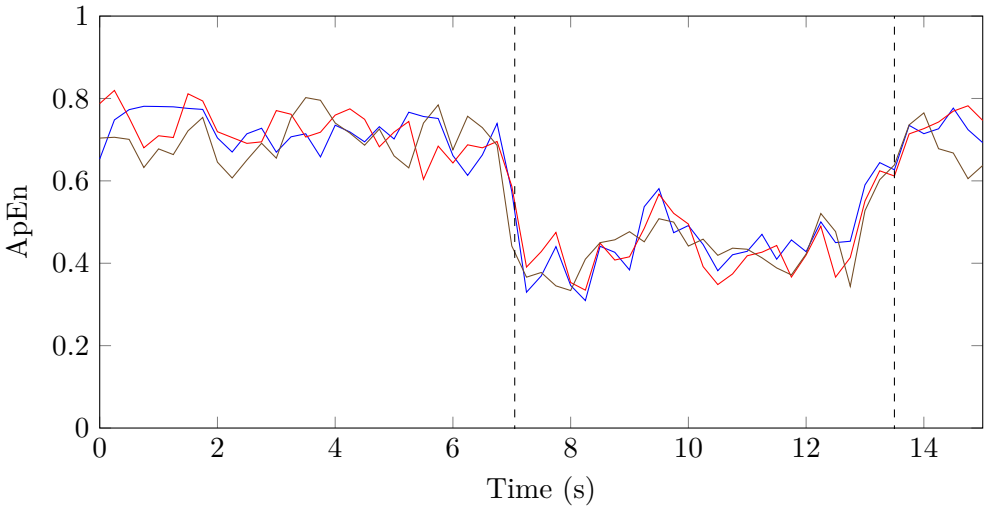
Figure 4.4: The power of DWT δ, θ, α and β bands

The variance is calculated on a sliding half overlapping window. The amount of overlap of each window is a trade-off between computational time and amount of resulting data points. The sliding-window step s can vary from $1 \cdots N$, when $s = N$, there is no overlap at all. We use $s = N/2$.

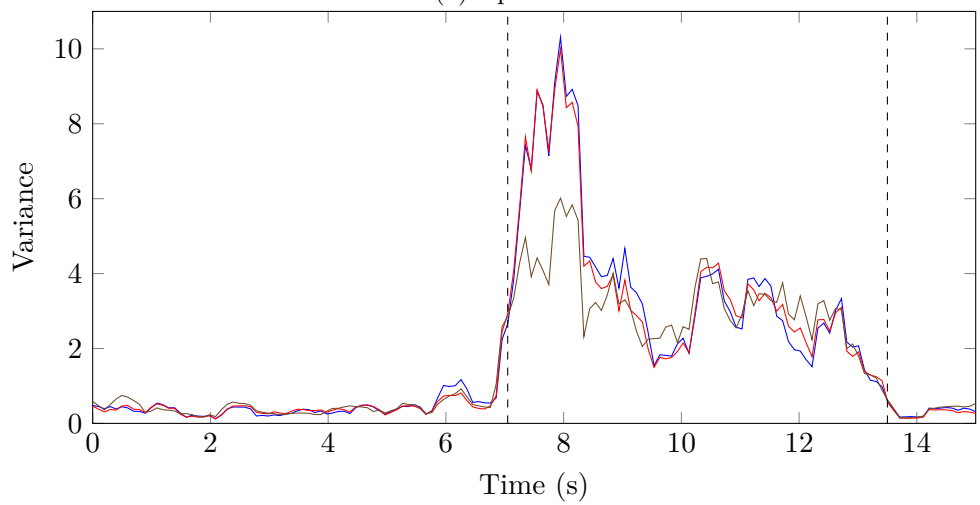
The variance of the ECoG signal vastly increases during a seizures as can be seen in Figure 4.5b. Also, during the inter-ictal and pre-ictal period, small but sudden peaks of variance are visible, which can possibly be an event to be learned from by a machine-learning algorithm.

4.3 Neural-network parameters

In this section, we will discuss the meta-parameters for the experiments, which are evaluated to find the best-performing combination. We will discuss the neural-network



(a) ApEn



(b) Variance

Figure 4.5: ApEn and variance for each of the 3 channels of normalised input data

types to be used, which activation functions to try, the number of hidden layers and their size, and the input and output formatting.

As we have seen, there are three types of neural networks that we will evaluate: the MLP, LSTM and WTTE-RNN. The latter has a defined choice for its output-layer activation functions and output data format, see subsection 2.5.6, which we will use.

There are several activation functions that are popular nowadays. The choice depends on the kind of problem, on whether it is a regression or classification problem, and on the layer type. Hidden layers usually have a different activation function than the output layer. For hidden layers the choices are multifold, common used functions are *sigmoid*, softplus and Leaky ReLu, which are the function we also will evaluate. For the output layer, the activation function is mainly dependent on the kind of prediction problem. For

multi-class classification problems, the softmax function is advisable, because the output values will have a cumulative value of 1. Each output value corresponds to the chance of the corresponding class being true, and selecting the highest output value yields the classified class. For regression, the output-layer activation function should be able to output the range of values we expect. Therefore the *tanh* or sigmoid functions are less usable as they have an output range of $[-1, 1]$ and $[0, 1]$ respectively. Variations of the ReLu activation function are very suitable because they can output any positive number, and we will therefore use the Leaky ReLu in our work with an α of 0.2, as deducted from Xu et al. [50].

Another type of functions is used to determine the loss of each training step. Every time a input sample is fed into the ANN it is used to predict an output. This output is then compared to the expected output, which is present in the training data along with the used input sample. The error between the predicted and expected output is then calculated using a loss function.

In this work two types of loss functions are used, one for classification and one for TTE prediction. For classification as loss function is needed which can determine the error of the predicted class-probabilities. The best suitable loss function is therefore the categorical cross-entropy loss function. For the WTTE-RNN a specialised loss function is used, as proposed by [20], the discrete Weibull log-likelihood function.

There is no analytical method for determining the number of layers and layer size. Therefore, we have to try multiple options and compare them to be able to select the best combination of options. To limit the potentially exploding amount of layer combinations we will limit ourselves to using layers of the same size only. In this work, we will explore ANNs with 1, 2 and 3 hidden layers. The more layers one uses, the more complex patterns can be learned. But having more layers makes it harder to learn and it takes more training epochs and training data to get to a good accuracy. Based on several tries we choose for 1, 2 and 3 layers. Each layer will have 10, 30, 90 or 120 neurons, these numbers are also based on multiple tries of running the network. When using more than 3 layers or more than 120 neurons per layer, we observed that the network did not converge within 200 epochs, even with a low learning rate, and that the needed computational power increased too much to be feasible to run on the available hardware.

The input window determines the amount of data that is fed to the network at once. For an MLP, this means that this is also the time window which it can use to learn from because it is a non-recurrent network. We will use input time windows of 0.5s, 1s, 3s, 5s and 10s. Larger time windows are hard to use because the network does not fit into memory anymore. Also, making the window much larger would mean that a window could include two seizures because the seizure interval in the data is on average 60 seconds. This would lead to training the network to predict the seizure based on the time of the last seizure, which is not the goal in this research. This also means that we are limited to a prediction time of 10s. But as the input-data contains a seizure about every minute, we expect any available precursor to be present in the selected input windows as well.

There are three ways of formatting the neural-network's output for the purposes of this work. The first is a one-hot encoded class for classification in combination with the softmax activation function. One-hot encoded means that the output is a bit-vector

with a single 1. The position of the 1 corresponds with the predicted class. The second is a single output value or array of values which are a prediction of the next values of the input signal. This can be used to evaluate if the network predicts anything sensible at all. The third option is used for the WTTE-RNN and outputs the two parameters of the Weibull distribution. These two parameters depict the time probability distribution until the next seizure.

Further, optimisers have been used to increase the overall learning accuracy. An optimiser algorithm optimises the way weights are adjusted in the network and can also alter the learning rate based on parameters of the network. The algorithm itself determines what parameters are changed and in what way. We have used a standard optimiser named RMSprop, which mainly optimises the learning rate.

Another parameter often used in neural networks are drop-out layers. These are layers in the network that randomly drop-out or ignore output parameters of neurons. This prevents the network from over-fitting: learning for too specific cases, so the network is not generalised enough anymore. We use a drop-out factor of 0.2 for all but the output layer, meaning that 20% of the neuron output values is dropped for these layers.

4.4 Output data post-processing

The output of the network needs to be post-processed and evaluated to determine the performance of the system. For the evaluation, we will use the annotations of the input data, which indicate the start and stop times of each seizure.

For classification, the output of the network is a one-hot encoded value, which has to be converted to an integer value. This is needed so it can be compared to the annotated class values and they are easier to save and plot. One-hot encoded means that instead of a normal integer value, a class is represented by a list of binary values, where the position of the only 1 in the list corresponds with the categorical class. The output of an ANN is non-binary, but provides a chance for each class. The highest value in the list is the most probable class, so the conversion can easily be done by selecting the highest value from the output list, and use the position of this value as class number. Post-processing also includes a K-nearest Neighbor (KNN) filter, to filter out jitter of the network and make for a more stable output signal. A KNN filter looks at a configurable number of neighbour data-points and determines its output based on the majority of the data-points with the same value.

When using sample prediction, we train the network to predict the input signal, and in case of multi-channel input, the average of the input channels. This does not predict seizures directly, but we can see if the network predicts values that correspond with the expected output, which is the input signal one time-step ahead. If the input data was scaled down, we need to scale the predicted output back up to get to the original value range.

The WTTE-RNN outputs the two parameters of the Weibull distribution and we will mainly look at the first parameter which depicts the time until the next event. If the system cannot predict a seizure, this parameter will be around the average value of the training times. That means that the network is not able to predict a value with high

certainty, but will output a generalised value, which is about the average of all possible outputs.

To evaluate the performance we will determine the following metrics:

- True Positive Rate (TPR): Also named sensitivity, is a value that depicts the percentage of correctly predicted seizures. TPR is defined as:

$$TPR = \frac{TP}{FN + TP} \tag{4.1}$$

  where true-positive (TP) is the number of correctly predicted seizures and false-negative (FN) is the number of undetected seizures while there really was a seizure.

- True Negative Rate (TNR): Also called specificity, describes the percentage of correctly classified inter-ictal periods, so the periods without seizure. A lot of false predictions make for a low TNR value. The TNR is defined as:

$$TNR = \frac{TN}{FP + TN} \tag{4.2}$$

  where true-negative (TN) are the number of truly predicted inter-ictal periods (with no seizures occurring), and false-positive (FP) is the number of falsely predicted seizures.

- Average Prediction Rate (APR): The average of TPR and TNR.

- Prediction time: The time between the prediction of the seizure and the actual seizure occurrence.

- Categorical cross-entropy loss (only for classification): The average loss over the last training epoch as calculated by the cross-entropy loss function. This gives an indication of whether the predicted output classes are a random guess or actual predictions.

Key for calculating the TPR, TNR, APR and prediction time is accurate detection of the beginning and end of the pre-ictal and ictal period. This differs for classification-output and when using the WTTE approach.

For a categorical class-output, we look at the pre-ictal and ictal classes, and their start- and end-times. If a pre-ictal class is present, it should be followed directly by an ictal class to be a positive prediction of a seizure. Otherwise, it is counted as false-negative. The prediction time is based on the start of the pre-ictal class, or when the pre-ictal class is not predicted before an ictal class, the start of the ictal class. The prediction time is then calculated by comparing it to the annotated start time of the seizure. A negative prediction time is the same as a detection delay.

For TTE output, this is a bit more difficult as the output is a continuous value. The start of the pre-ictal or ictal period can be identified by a decreasing output value, because when the predicted time until the next event decreases, this actually means that a upcoming seizure is predicted. During inter-ictal periods, the output is more or less stable and converged to an average value of training values. We define the start of the

pre-ictal period as the moment when the predicted time is significantly lower, meaning at least 20% lower, than the value during the ictal period. The threshold value of 20% has been empirically determined by looking at the variation of the output signal.

Categorical cross-entropy loss is calculated using the predicted class outputs and is used during training to adjust the weights of each neuron. This output consists out of the normalised probabilities for each possible class. The loss is then calculated based on these probabilities and the expected output class using the following formula:

$$f(x) = -\sum_{c=1}^{M} y_{o,c} log(p_{o,c}) \tag{4.3}$$

where M is the number of possible classes, y a binary indicator if class c is correct for observation o and p is the predicted probability of observation o for class c. The Categorical cross-entropy loss gives an indication of whether the predicted output classes are a random guess or actual predictions based on the input data.

4.5 Software architecture

The neural networks are implemented in Python using the framework Keras [51]. Keras is a framework designed for instantiating model-based neural networks and depends on a low-level, high-performance back-end computing library. Multiple of these back-end libraries are supported. The back-end library we use is TensorFlow [52], because it is widely used, is still under active development, works stably and has support for running on one or more GPUs.

The software implemented for the experiments of this research consists out of several independent parts, see Figure 4.6 for how they relate to each other. The pre-processor is used to fetch the data from the recording files and also for generating the training samples used by the neural network. The analyser part is used for calculating the desired features, and the Trainer and Validator part are used to perform the learning and evaluation.

The pre-processor reads the data from the recording files and saves the data to a better-structured and more easily processable file. The original recordings are available as ABF files, where ABF stands for Axon Binary File, a file format used by the recording devices from Molecular Devices [53]. In the pre-processing step, the data from these files is read using the Python library Neo [54], and converted to a Matlab file with a certain structure reflecting the Neo objects. Also the Excel files with annotation data are converted to a list of begin- and end-times of the seizures of the corresponding ABF file, and added to the Matlab file so they are easily usable and accessible when creating the training samples. So, for each source file a destination file is created including the recording and annotation data. This process has to be done only once.

The generation of training samples is a separate process and has to be done for each different type of experiment. The generator generates training samples, that is, a collection of input and output values for the supervised learning algorithm. To generate these the following parameters are used:

- **Channels**: The number of channels in the training samples. For source files with

more than the configured number of channels, one or more random channels are discarded. Source files with less than the required number of channels are ignored.

- **Input window**: The width of the window included in each input sample in seconds. The number of values in each input sample is calculated based on the sampling frequency of the signal.
- **Input step**: The step in seconds between input windows. So e.g., if the step is half the input window, samples are half-overlapping.
- **Input sample rate**: The sampling frequency of the input samples.
- **Output window**: The width of the output sample window.
- **Output sample rate**: The sampling frequency of the output samples.
- **Output offset**: The offset of the output relative to the input. This is used to be able to make an ANN predict values in the future, e.g., by using an offset equal to the input window.
- **Output type**: The type of output, can be either one-hot encoded classes, input values or values used for the WTTE-RNN network. Each output type has additional parameters which can be specified, e.g., the duration of each class for a categorical output.
- **Pre-processor**: A reference to a function can be passed to the generation class which pre-processes the input data. In this function, features can be calculated from the normalised input data, which is passed as parameter to the function.

The generated samples are saved in files with a certain amount of samples to limit the file size.

The analysis package can be used to process the input data and calculate different features. It supports calculating averages, scaling the data to a certain variance, detecting edges and can calculate event frequencies. Further, it has functionality for performing mathematical operations on the whole time-series data-objects, which are then performed on each value in the data-object. For example, in case of subtracting two time-series data-objects, each value at the same index in the objects will be subtracted, resulting in a new data-object. Also adding, subtracting and multiplying with a scalar is supported. Other parts of the analysis package allow for calculating correlations, variance, entropy, energy and applying wavelet filters.

The next part of our application is the machine-learner package which consists of a base model from which the different neural networks are derived: MLP, LSTM and WTTE-RNN. The models are used for training an validation of the ANN and are based on Keras. To be able to find the optimal meta-parameters, a grid-search module has been implemented which tries all combinations of a certain set of parameters.

The plotter package is used for plotting graphs and it uses a little webserver so graphs can be viewed in the browser. The loaded page shows the graphs in chunks and allows for navigating through these chunks. This is more convenient because the time span of the plotted data is in general too long to be plotted in a single graph.

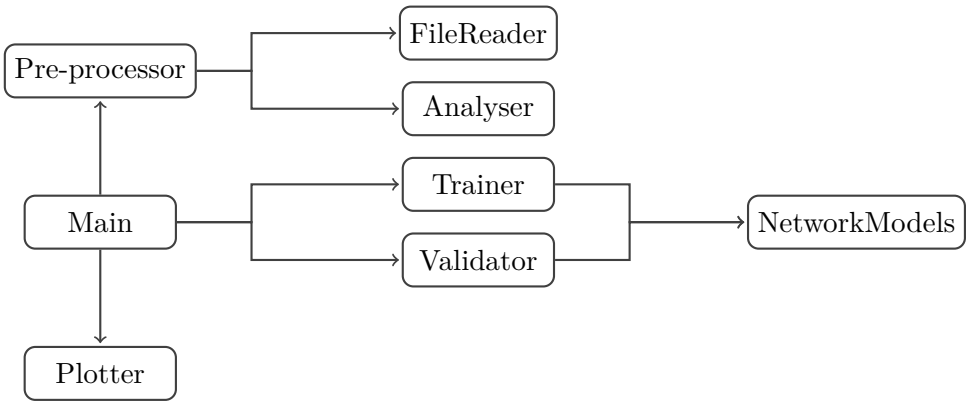


Figure 4.6: Diagram of the modules in the Python application.

5 Evaluation

In this chapter, the results of the different experiments with varying neural-network configurations are presented and evaluated. The performance of different configurations is compared in terms of prediction time and Average Prediction Rate (APR). First, we will show the set-up for the experiments, after which the evaluation criteria and results are set out. The results are divided in two parts, where the first part discusses experiments with an input set containing the normalised recordings and the second part discusses experiments with generated features as input. Also we will report some exploratory results on using Time-To-Event (TTE) prediction outputs. Finally the results will be discussed and evaluated.

5.1 Electrocorticography (ECoG) recordings

The data used for the research in this thesis are ECoG recordings from the Erasmus Medical Center in Rotterdam, recorded at the Neuroscience department. The data is recorded on male and female natural mutant tg mice, ranging from 4- to 30-weeks old, which have naturally occurring absence seizures. Also 8- to 10-week-old inbred C3H/HeOuJ mice with natural-occurring absence seizures have been used. Both types wer bred using heterozygous parents. See [4] for a more detailed description of the mouse colony.

The ECoG recording electrodes were placed directly on the brain under the scalp. The recordings were originally done for a research thread where they inject the mice with either a muscimol solution, to decrease the neuronal activity, or with gabazine solution, to increase the neural activity [4]. The goal was to to determine the relation between the injection and seizure frequency and duration. The ECoG recordings were filtered online using a 1-to-100Hz band pass filter, and a 50Hz notch filter. We only use data recorded before the injections, so the recordings are not influenced by the injected solutions, but contain only natural occurring seizures.

The recorded signals were manually annotated by employees of the neuroscience department which are experts in identifying absence seizures. Only channels with good recordings were selected, which means that channels which did not contain proper recordings, e.g. no signal at all or just noise, were discarded. The resulting dataset has therefore recordings with one, two or three channels of ECoG data, see Table 5.1. When running experiments where we only need one or two channels while the input data has more channels, the remaining channels are just ignored and not used. The part of data that has less channels than required is also ignored.

The total amount of recordings is 116.87 hours and the data contains a total of 7,371 seizures. This means that the data contains on average 63.1 seizures per hour.

Number of channels	Hours of recordings	Percentage of total
1	83.54 h	71.5 %
2	3.86 h	3.3 %
3	29.47 h	25.2 %

Table 5.1: Amount of recorded data

Parameter	Values				
Network type	MLP	LSTM	GRU		
Input type	3 channels normalised data, selected features				
Output	Inter-ictal, pre-ictal and ictal classes				
Number of channels	3				
Input window	0.5s	1s	3s	5s	10s
Input step	0.25s	0.5s	1.5s	2.5s	5s
Number of layers	1	2	3		
Number of neurons per layer	10	30	90	120	
Pre-ictal time	5s	10s	20s		
Number of epochs	50	100	150		
Hidden layer activation function	Sigmoid	Softplus	LReLu ($\alpha = 0.2$)		
Output layer activation function	Softmax				
Loss function	Categorical Cross-Entropy				

Table 5.2: Overview of classification parameters

5.2 Experimental set-up

The experiments are run on 29.47 hours of data with recordings of three channels. More single-channel data is also available, but it is believed that the relation between several recording channels will improve the results. It has been shown that changes in synchronicity between multiple brain areas can be an indication of a pre-ictal state, and it is also known that the Spike-and-Wave Discharges (SWDs) spikes spread over the whole brain [55]. After all, three channels contain more usable data than just one. There is almost no two-channel data available in our data set, so it would make less sense to choose for 2-channel training data.

The implemented software used for the experiments is written in Python and is therefore cross-platform. The experiments have been run on a desktop computer with Windows 10 and a server with Ubuntu 16.04. The computational performance of the application will not be evaluated as it is not the goal to optimise this.

For each experiment we use a range of values for each input parameter, of which all combinations will be tried. The goal is to find the optimal combination of parameters. All parameters are further explained in section 4.3 (page 36).

The parameters are listed in Table 5.2 and Table 5.6 for classification and TTE prediction respectively. Also the evaluation is different, as calculating the accuracy of

Parameter	Values
Network type	GRU
Input type	3 channels normalised data
Output	Weibull distribution parameters λ and k
Number of channels	3
Input window	0.5s
Input step	0.033s
Number of layers	1
Number of neurons per layer	5
Pre-ictal time	4s
Number of epochs	75
Hidden layer activation function	Tan-h
Output layer activation function	Exponential (α neuron) and soft-plus (β neuron)
Loss function	Log-likelihood

Table 5.3: Overview of the TTE model input parameters

a predicted class differs from determining the accuracy of a decreasing time prediction, further explained in the next section.

The total data-set is split into three parts: one for training, one for evaluation during training and one for testing the trained model. The training data-set contains 72% of all samples, the evaluation data-set 8% and the test data-set the remaining 20%. The evaluation data-set is used during training to evaluate the training process, and the reported loss is determined during this evaluation process. The test data-set is only used after the model was trained, to verify the performance and to determine the metrics as described in section 5.3.

The input window- and step-times have been chosen based on initial empirical try-outs and the same counts for the network size. Also for the number epochs we looked at the try-out results and especially at when the change in reported accuracy became very low. The loss function and output-layer activation functions are specific for classification purposes, while for the hidden-layer activation function we will cover three of the most popular ones.

5.3 Evaluation criteria

The prediction performance is evaluated differently for classification and TTE prediction, as further explained in the next sections. The evaluation for both types is done in terms of the following metrics:

- **True Positive Rate (TPR)**
- **True Negative Rate (TNR)**
- **Average Prediction Rate (APR)**

- **Prediction time**
- **Categorical cross-entropy loss**

See section 4.4: Output data post-processing for more details on the implementation of these metrics.

5.3.1 Classification evaluation

For the performance evaluation of the class-output we will look at the predicted pre-ictal and ictal classes and their timings. The other output class, the inter-ictal class, will be ignored as it is the default class and does not predict or identify a seizure in any way apart from the absence of a seizure. The class output is represented by a integer value, and the output of the neural network is post processed as described in section 4.4.

A predicted pre-ictal class is only valid when it is directly followed by a ictal-class. Otherwise it means that the system first predicted a seizure after which it withdraws this prediction. If a pre-ictal class is not followed by a ictal-class, it counts as a false positive. Also a predicted ictal class without actual seizure in the data is counted as a false positive. If no seizure is predicted or detected when there is a seizure in the signal, it is counted as false negative.

Whether an ictal class is counted as a correct seizure detection or not is depending on the amount of overlap with the actual seizure. We define that a seizure is detected correctly if the ictal-class overlaps for at least 50% with the annotated seizure. If the ictal-class is starting before the actual seizure, it adds to the prediction time. But if it is lagging behind, this either means a negative prediction time in the case that no pre-ictal-class was present before the ictal-class, or that the pre-ictal class partially overlaps with the actual seizure. For the latter, we still use the time difference between the start of the pre-ictal-class and the annotated start of the seizure as prediction time.

If the system detects multiple separate seizures within the time of one particular seizure it is counted as one true positive. This can be the case if the output contains jitter, e.g. when within the seizure an inter-ictal class is predicted for a short period of time.

The prediction time is the time between the start of a seizure prediction and the actual occurrence of the seizure. It is determined by the start time of the pre-ictal class, and if it is absent, the start of the ictal class, compared to the start of the actual seizure which is available through annotations of the input data. If the pre-ictal class is not present in the output, but the ictal class is, we are actually detecting the seizure instead of predicting it, and the prediction time will be close to, or below zero.

The performance will be evaluated using the TPR, TNR, APR, prediction time and categorical cross-entropy loss metrics, as further explained in section 4.4.

5.3.2 Time-To-Event prediction evaluation

Using the Weibull distribution output parameters to predict the TTE makes for a different evaluation method than when evaluating classification outputs. The predicted time until the next event is a continuous number which should decrease when a seizure is

upcoming. The system is trained with a time until the next event which is equal to the time to the next seizure. Only if the time to the next seizure is larger than the pre-ictal time, the pre-ictal time is used, so we have a maximum TTE value with which we train the network.

$$f(x) = \begin{cases} TTE, & \text{if } TTE < PIT \\ PIT, & \text{otherwise} \end{cases} \tag{5.1}$$

where TTE is the time to the next event, and PIT is the chosen pre-ictal time. During an ictal period the TTE is trained to be 0.

When no event is predicted by the system, it will output a value around the pre-ictal time, and when a seizure is predicted, this value will decrease. To prevent false-positives we use a threshold of 20% to determine when the start of a prediction is, so when the predicted time to the next seizure drops below 80% of the configured pre-ictal time, we say that a seizure is predicted.

A seizure prediction is only valid when the prediction is continuous and not cancelled. The prediction is cancelled when the predicted time goes up again to a value above 80% of the chosen pre-ictal time.

For the determination of the prediction time, which is the time the system can predict a seizure in advance, the same counts as for classification output. If the prediction time is zero or lower, the prediction is rather a detection, but we will still report it as prediction time which will then be negative. Also the calculation of the metrics is done in the same way as for class-prediction, apart from the fact that the pre-ictal and ictal starting times are now derived from the TTE metric instead of from the class outputs.

5.4 Prediction performance

In this section we present the prediction performance using the neural network model. First we describe the classification performance using the normalised input and using generated features based on the input data. Secondly we present some exploratory results of a recurrent neural network which can predict times-to-events based on the Weibull distribution.

5.4.1 Classification performance

For each experiment a neural network is set up with several varying input parameters. The parameters and the different used values are described in section 5.2. Because one cannot know the exact influence of each parameter on the performance of the neural network, we did an extensive grid search, meaning that an experiment has been run for every combination of input parameters.

The amount of combinations had to be limited to be able to run one whole grid search for one set of data, either the normalised input data or the generated features, within a week. Therefore, the number of varying parameters is limited to four, with each three or four different input values resulting in a total number of experiments for each data set of 108. The varied input parameters are the network type, the input window and the network size, consisting of the layer size and number of layers.

The output of the model is a prediction of the during training learned output values. In our case of using a classifier, the output values are one of the learned classes. Based on these predicted classes several metrics can be calculated by comparing them to the expected and annotated output. The metrics we use for presenting the performance results are the APR, TNR, TPR and the prediction time. How they are determined is further described in the previous section, section 5.3. We also use the categorical cross-entropy loss as performance metric. This metric is not based on the output values of the neural network evaluated using a test data-set, but is rather a neural network parameter used during training to evaluate the performance and is determined based on the evaluation data-set every training epoch. The evaluation data-set is used for determining the loss value, because it is already available during training, it is used by the network to update the weights and not based on the training data-set.

To be able to compare the varying input parameters against the resulting output parameters, we have selected several combinations which will be presented for both the experiments ran on the normalised data and the generated features.

The first comparison will be made by varying the input window for all three network types and plotting the resulting APR. This shows the influence of the amount of input data on the performance of the network, and makes the results for each network type comparable.

Then, we will show the average APR, TNR and TPR of runs of all network types for varying input windows to be able to compare the difference in number of true-positives and number of false-negatives. The TNR depicts the relative amount of false-negatives and the TPR the amount of true-positives.

In the third and fourth graph we will plot the resulting average APR and categorical cross-entropy loss of multiple window sizes, for the used layer sizes. This will give insight into the relation between the window size and the layer size regarding the seizure prediction performance and network loss.

Lastly, the average APR and categorical cross-entropy loss will be displayed for the used layer sizes and amount of layers, to be able to determine the best combination of those two parameter.

5.4.1.1 Results using normalised input

In this section we will look at the results using the normalised input. This input data is the raw measured ECoG data, but then resampled to 300Hz, scaled to unit variance and normalised to having a zero mean. The data will then look as shown in Figure 4.2b. See subsection 4.2.1 for a detailed description on the normalisation process.

First we will look at the results of a varying input window. The input window determines the amount of data fed to the network each step and determines the size of the first layer, also called the input layer. The more data is fed into the network each time step, the more information is available, possibly making for a better prediction performance. But on the other hand it is harder to relate data to each other and learn from the input data because the relations that have to be learned become more and more complex. Also, certain unwanted and unrecognised artifacts of pre-processing the input-data can get a higher influence on the learning process.

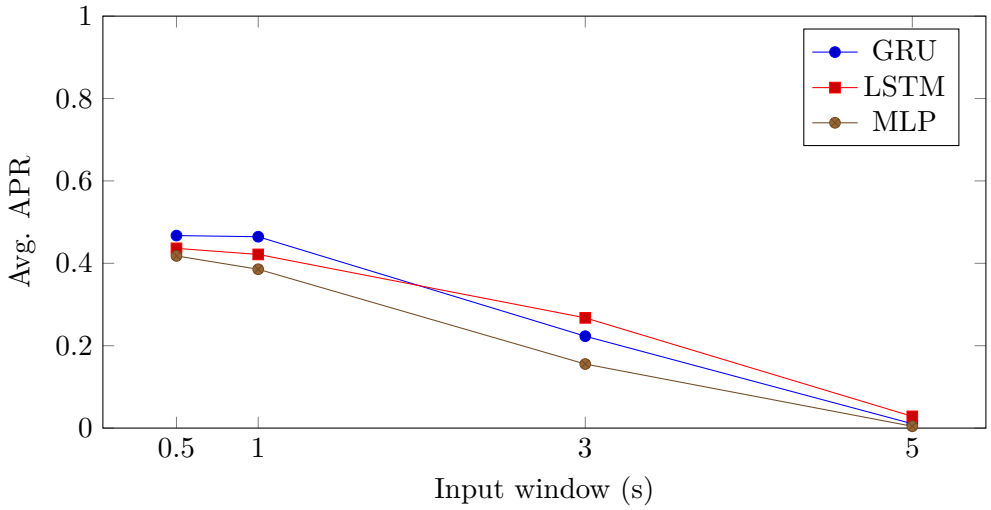


Figure 5.1: Resulting Avg. APR for multiple input windows and network types

In Figure 5.1 we see the average resulting APR for several input windows and network types. The results for each different network type are close to each other. The average APR however decreases with an increasing input window size. If we look at the differences between the TPR and the TNR (Figure 5.2) we see that the TPR is always higher than the TNR, which means that the true seizures are being detected, but there are also a lot of false positives making for to a low TNR. This is also shown in the bar graph (Figure 5.3) which shows the avg. true-positive (TP), false-positive (FP), true-negative (TN) and false-negative (FN) counts for different input windows. The number of false positives is a lot higher than the other counts, which supports that the number of false-positives makes for a low APR. Also, if we compare a small window to a larger window, the number of TPs and TNs decreases, while mainly the number of FNs increases. This means that for larger windows there are a lot more cases that the network predicts the inter-ictal class, while the output should be a pre-ictal or ictal class.

In Figure 5.4 and Figure 5.5 we will look at the network size and its influence on the prediction performance expressed in the average APR and categorical cross-entropy loss. The first of the two figures shows the avg. APR for a number of input windows and different layer sizes, and the second figure shows the categorical cross-entropy loss for the same parameters. Again the avg. APR decreases with a increasing window size no matter the layer size. But also the cross-entropy loss decreases meaning that the prediction error decreases. This indicates that the higher APR value for smaller windows is more due to random output values than due to actual predictions.

In Figure 5.6 and Figure 5.7 again the network size is compared with the average APR and categorical cross-entropy loss. This time we look at the number of layers and the layer size. The first of the two plots shows the avg. APR for a number of layer sizes and number of layers. There is no clear winner, all combinations seem to perform about the same. The second graph shows the cross-entropy loss which decreases a little for an increasing window size with the same trend for all amounts of layers. This indicates that

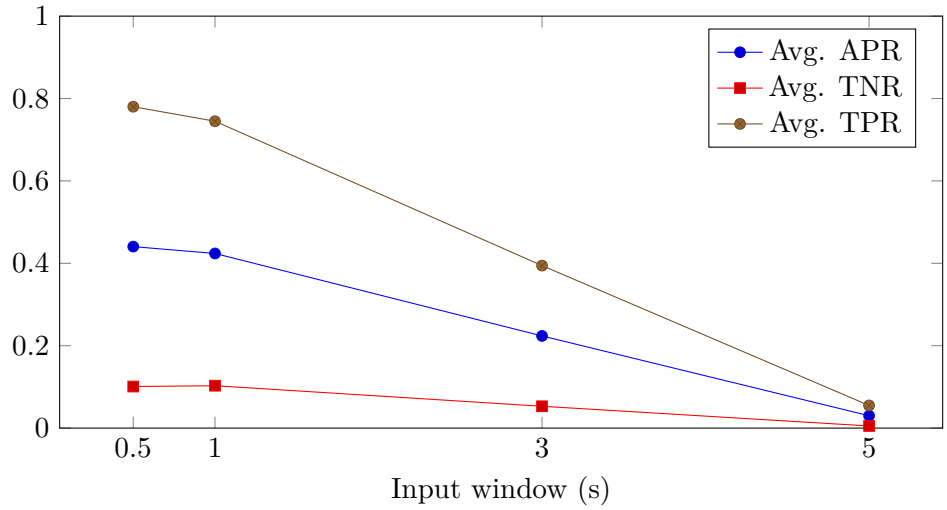
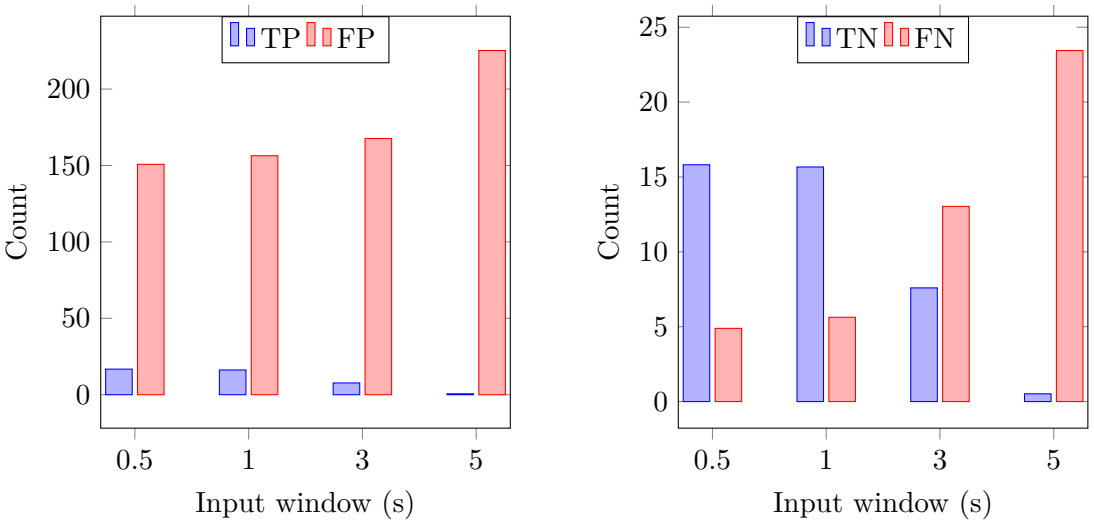


Figure 5.2: Resulting Avg. APR/TNR/TPR for multiple input windows



(a) TP and FP

(b) TN and FN

Figure 5.3: Resulting avg. TP, TN, FP, FN counts for different input windows

larger layers can actually improve the prediction performance, but as the APR does not become higher for larger layers, this does not work for our input data.

Looking at the prediction time, as shown in Figure 5.8, we see a clear optimal input window. Around an input window of 3 seconds, the prediction time is the highest, while other windows yield a lower prediction window. When we compare the different number of layers used in running the experiments, a higher number of layers has a slight advantage over fewer layers. One outlier is the result with an input window of 0.5s and 2 layers of neurons in the neural network. Here the result is lower than expected, with an average prediction time of 1s. This can be due to a suboptimal initial state of

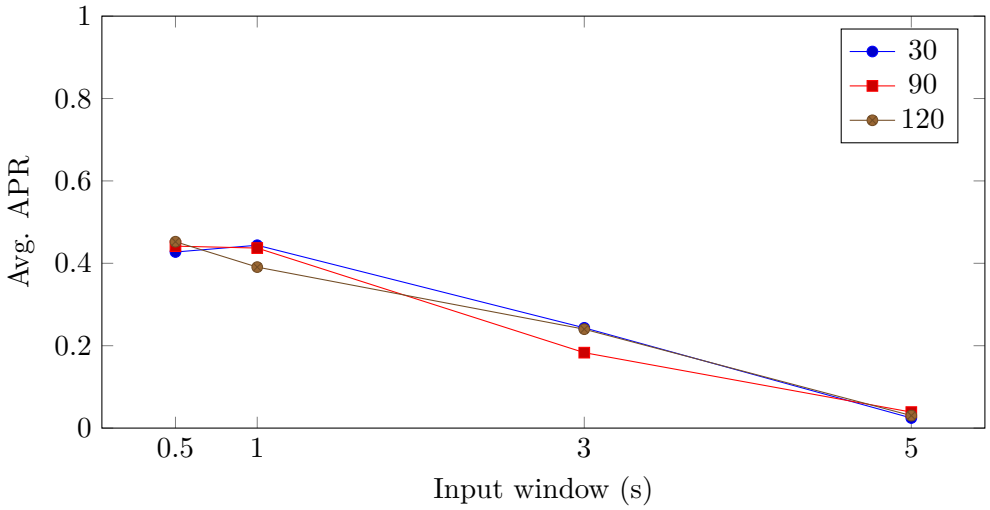


Figure 5.4: Resulting Avg. APR for multiple input windows and first layer sizes

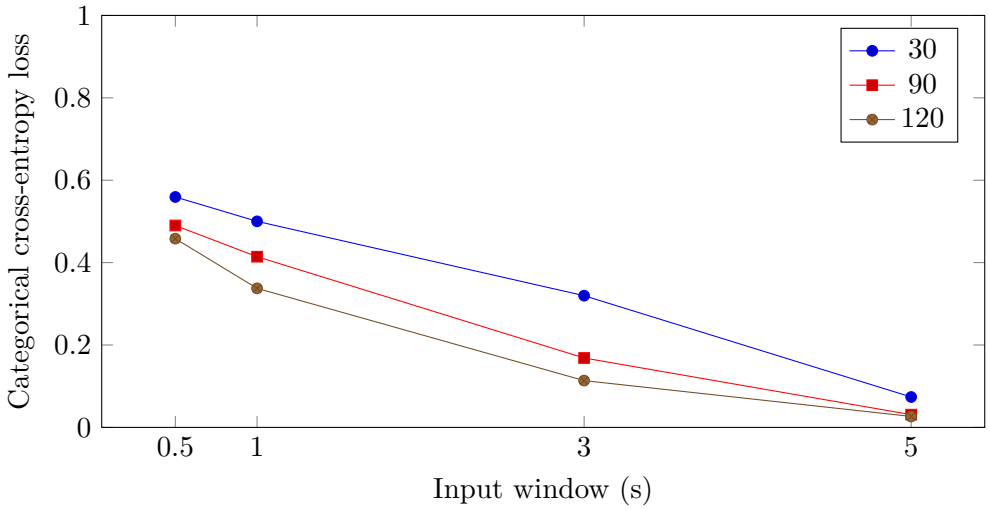


Figure 5.5: Resulting avg. categorical cross-entropy loss over the last training epoch for multiple input windows and first layer sizes

the neural network, as initial neuron thresholds and connection weights are initialised randomly. The influence of the initial state can be minimised by training the network multiple times and taking an average of the performance metrics. However, due to the computing complexity of the whole setup this would take too much time, which is why we have run the experiments only once.

Another risk of an increasing number of layers in the neural network is that it is easier for the network to overfit. We have tried to prevent this by using drop-out layers with a drop-out of 20% and by balancing the classes in the training-data, see also section 4.3 and subsection 4.2.1.

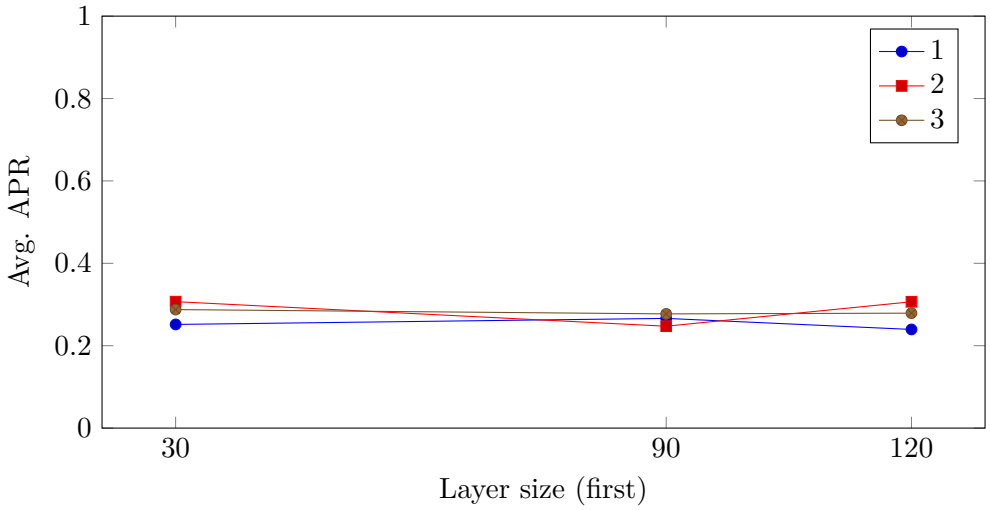


Figure 5.6: Resulting Avg. APR for multiple first layer sizes and amounts of layers

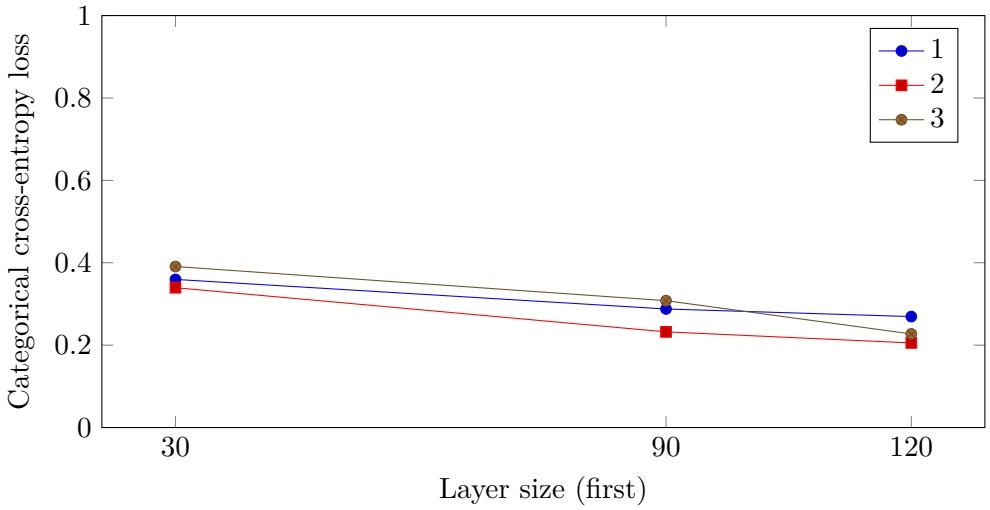


Figure 5.7: Resulting avg. categorical cross-entropy loss over the last training epoch for multiple first layer sizes and amounts of layers

The best performing combination of input parameters is listed in Table 5.4. These results are exemplary for the other results as presented above, and are characterised by a high TPR, but a very poor TNR. This means that seizures themselves are being detected, but also that during the time without seizures, a lot of false-positives occur. This might also be the reason for the relatively high prediction time of more than 3s, while the learned prediction time is only 5s. The predicted pre-ictal periods probably have been false-positives, like the many others that determine the TNR's low value. This is backed by the relatively low loss of 0.57, indicating that the output class for quite a number of input samples was a guess. However, the loss is significantly higher than for

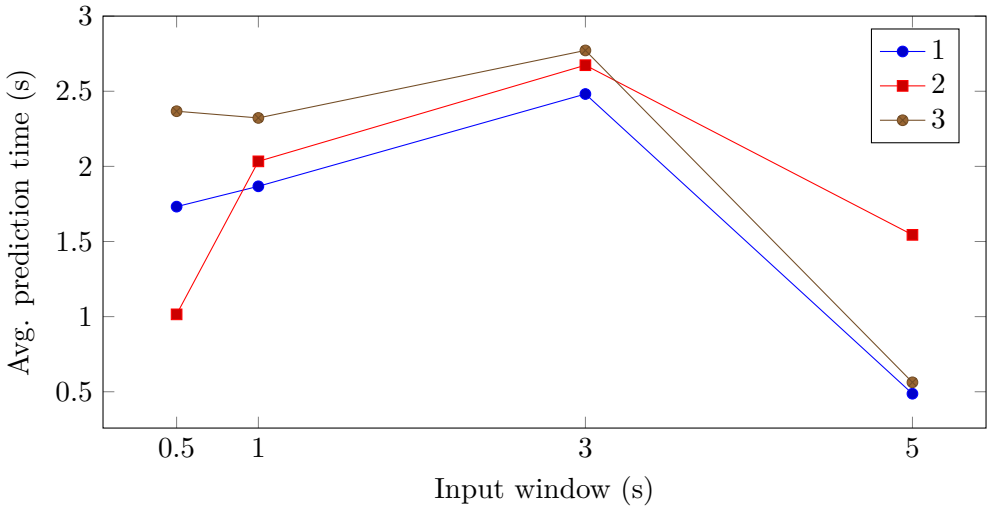


Figure 5.8: Resulting avg. prediction time for multiple input windows and layers counts

Input parameter	Value
Number of channels	3
Input window	0.5s
Input step	0.25s
Number of layers	3
Number of neurons per layer	30, 15, 5
Pre-ictal time	5s
Number of epochs	76
Hidden layer activation function	LReLu ($\alpha = 0.2$)
Output layer activation function	Softmax
Loss function	Categorical loss
Output metric	**Value**
Average Prediction Rate	0.57
True Positive Rate	1.00
True Negative Rate	0.13
Prediction time	3.11s
Categorical cross-entropy loss	0.57

Table 5.4: Metrics of best run using 3 channels normalised data set

true random output, because then the loss would have been around 0.33 for a tree-class classifier. See Figure 5.9 for a sample of the networks output of the best run, together with the input used to predict the output.

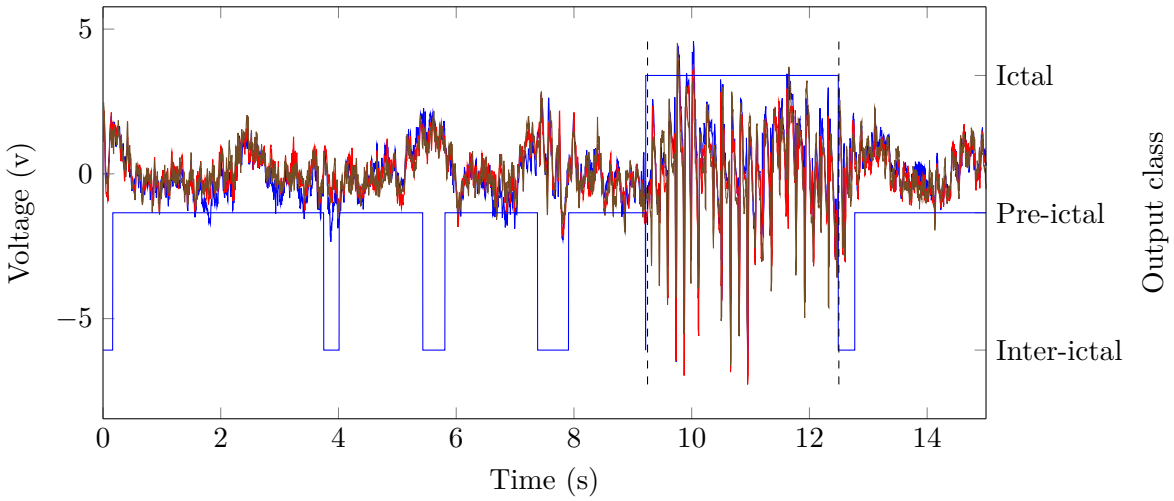


Figure 5.9: A sample of the predicted class output of the run with highest ADR

5.4.1.2 Results using selected features

In this section we will present plots with the same variables as in the previous section (subsubsection 5.4.1.1) but the results are from experiments with a different input set. While in the previous section we used normalised ECoG measurements, in this section we use a variety of features which are generated from the input data. In the next section we will compare the results of the two data sets.

The generated features used as input for our neural networks are the power of four frequency bands of Discrete-Wavelet Transform (DWT) signals, the variance of the signal and the Approximate Entropy (ApEn) for each channel, all generated from the normalised input data-set. The resulting amount of features for each time-step is therefore 18. The generation of these features is further described in section 4.2.

In Figure 5.10 the average APR is shown for GRU, LSTM and MLP types of neural network, for varying window sizes. Except when using an LSTM, the resulting APR decreases when increasing the input window. For the LSTM network there is an optimum for an input window of 1s. Figure 5.11 shows for the same experiments the averages of the APR, TNR and TPR to indicate the numbers contributing to the final APR, which is the average of the TNR and TPR. Clearly is visible that the TPR is a lot higher than the TNR meaning that seizures are being detected, but a lot of false-positives spoil the result. This is supported by the bar graph (Figure 5.12) showing the avg. TP, TN, FP and FN counts for different input windows. The number of false-cases (FP and FN) increase with an increasing windows size, while the number of true-cases (TP and TN) decrease. Especially for negative-cases (TN and FN) we see a large difference between a window of 0.5s and 5s. This shows that the number of wrongly predicted inter-ictal classes increases with an increasing window size.

The next two figures, Figure 5.13 and Figure 5.14, show the influence of the network size on the resulting APR and categorical cross-entropy loss. Again, an increasing window size results in a decreasing APR, except for a layer size of 90. The loss how-

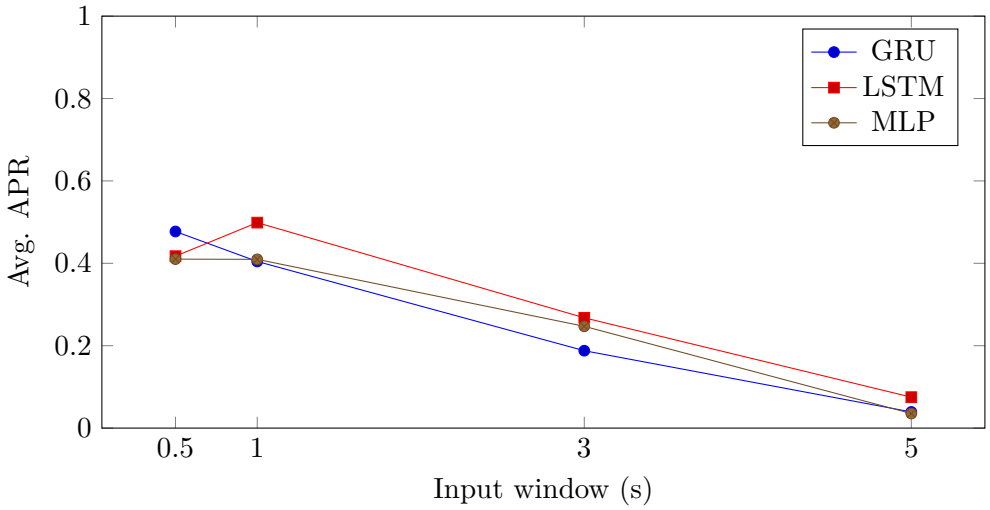


Figure 5.10: Resulting Avg. APR for multiple input windows and network types

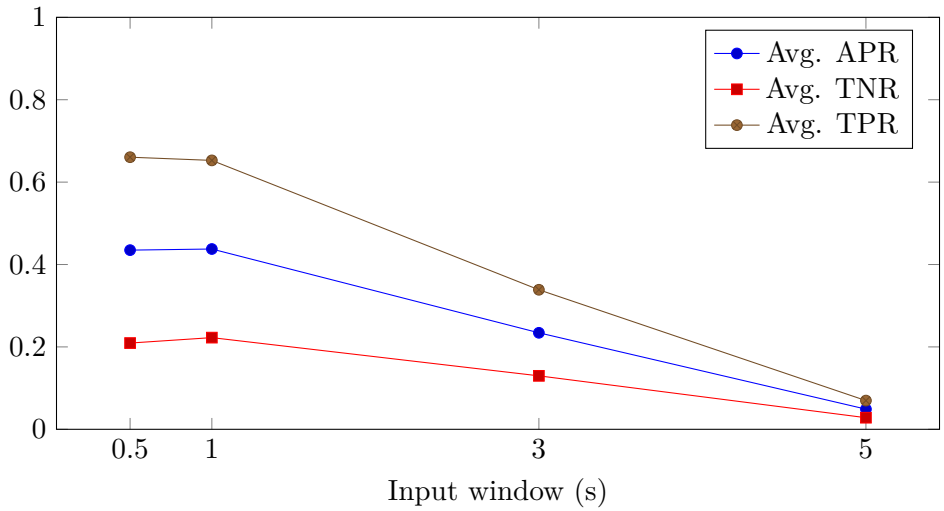
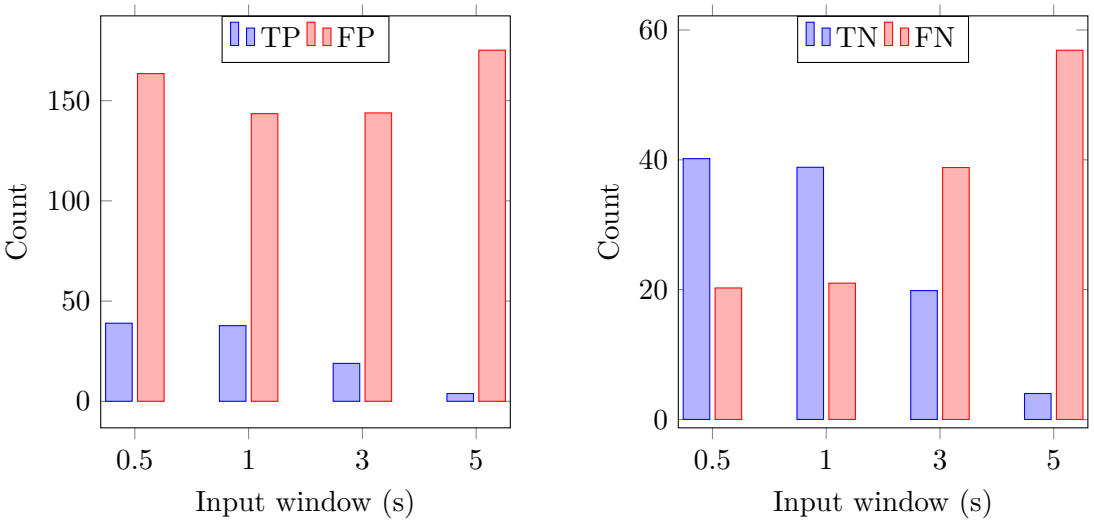


Figure 5.11: Resulting Avg. APR/TNR/TPR for multiple input windows

ever decreases for larger input windows and there is also a clear difference between the different layer sizes. The smaller the input window and layer size, the higher the loss becomes.

The following two graphs, Figure 5.15 and Figure 5.16 show the resulting average APR and categorical cross-entropy loss for varying layer sizes and number of layers. When looking at the average APR, there is almost no difference for the shown combinations of layer size and number of layers. The loss however, declines from about 0.4 to 0.2 when increasing the layer size from 30 to 120. This holds for all listed amounts of layers.

In Figure 5.17 the prediction times are plotted, showing us that with an input window



(a) TP and FP

(b) TN and FN

Figure 5.12: Resulting avg. TP, TN, FP, FN counts for different input windows

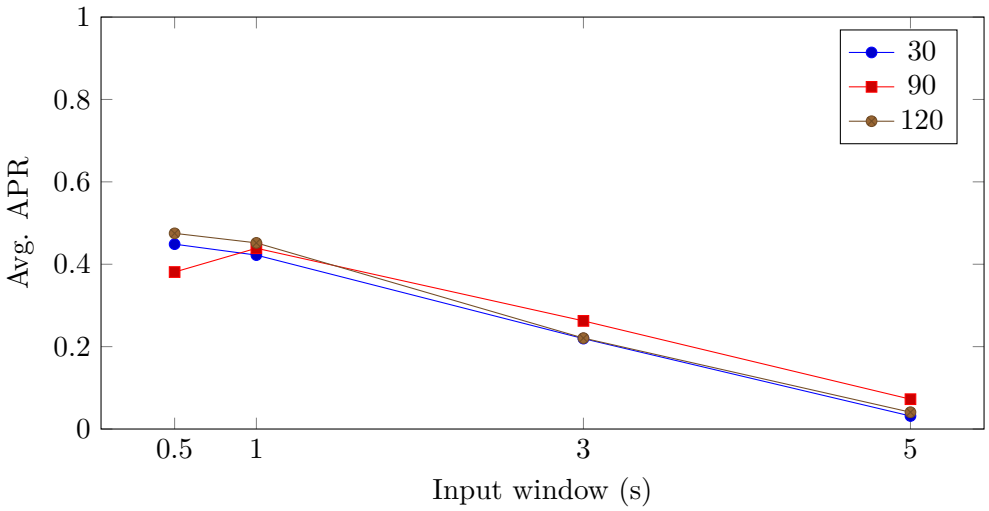


Figure 5.13: Resulting Avg. APR for multiple input windows and first layer sizes

of 3s the prediction time is the largest. The results are merely the same as with using the normalised input, and therefore shows that 3s of input data yields the best prediction times. These results are not related to the actual number of samples fed into the neural network each time step, as the input sample rate of the input set containing the generated features is only 100Hz, which is three times lower than the normalised input set.

The input parameters and output metrics of the best run using the features as input is listed in Table 5.5. A sample of input and output data is shown in Figure 5.18. Some of the features are left out of the plot because of the large total amount of features. For each channel the two lowest frequency power bands are left out, they became invisible

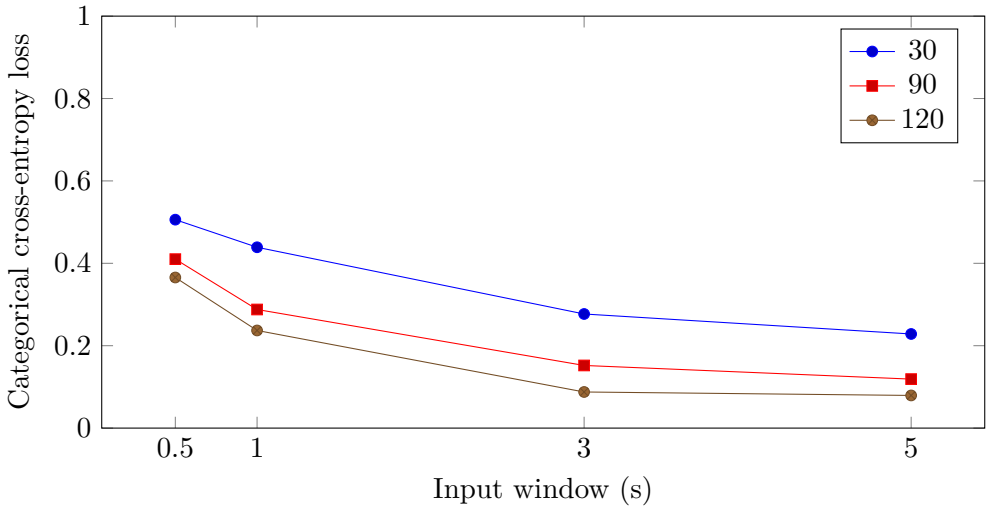


Figure 5.14: Resulting avg. categorical cross-entropy loss over the last training epoch for multiple input windows and first layer sizes

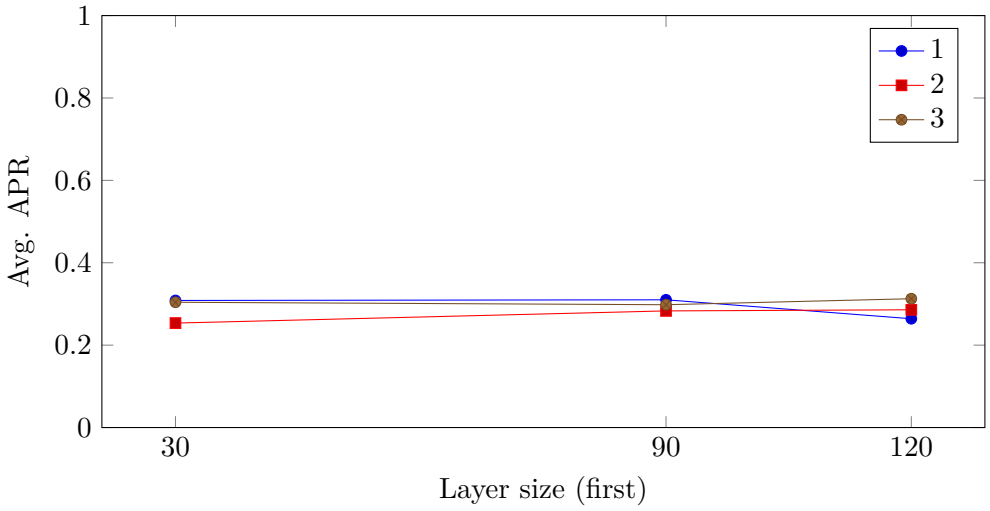


Figure 5.15: Resulting Avg. APR for multiple first layer sizes and amounts of layers

due to the other lines in the graph.

This experiment is exemplary for the results of the input set currently under investigation. The seizure itself is detected decently, even a bit ahead of the annotated seizure. This is the reason for the high average TPR. The inter-ictal and pre-ictal periods however, show a lot of false positives, leading to a low TNR. Whether the pre-ictal prediction just before the seizure is a random false-positive or an actual pre-ictal period prediction is hard to say. It does lead to a higher prediction time though, as all pre-ictal prediction that are just before a predicted ical-class add to the prediction time, which can therefore be misleadingly high.

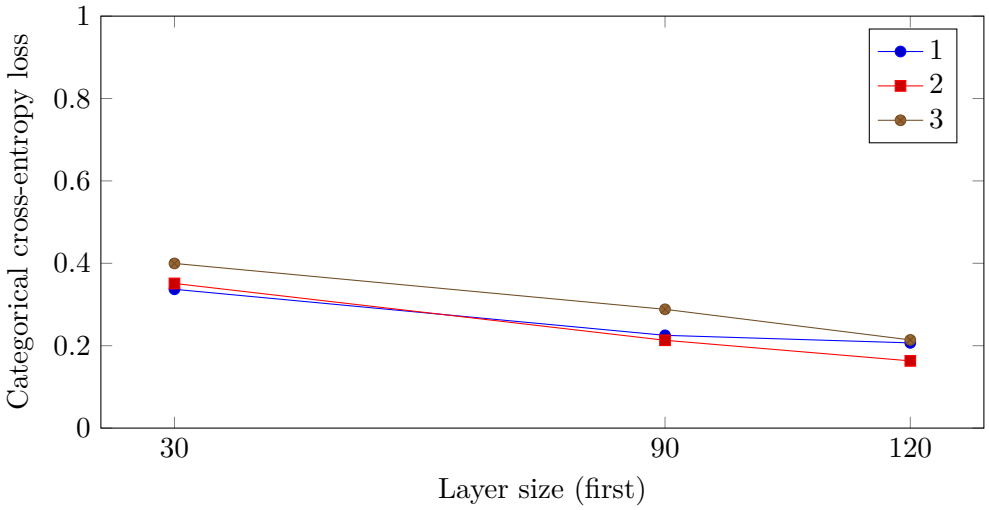


Figure 5.16: Resulting avg. categorical cross-entropy loss over the last training epoch for multiple first layer sizes and amounts of layers

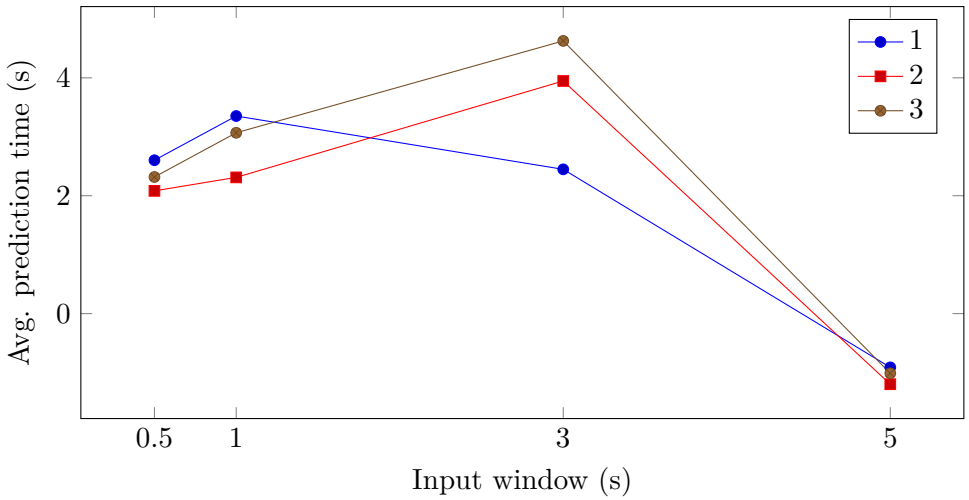


Figure 5.17: Resulting avg. prediction time for multiple input windows and layers counts

5.4.1.3 Comparison of input sets

In this section a comparison will be made between the two input sets as presented in the last two sections. The comparison is made between the average APR, TNR and TPR, see Figure 5.19. The red solid lines depict the results when using the normalised input, as described in subsubsection 5.4.1.1 and the blue dashed lines show us the results when using the generated features as input data for the neural network, as shown in subsubsection 5.4.1.2.

The average APR is about the same for both data sets, but the TPR and TNR differ. Where for the normalised input the TPR is higher, also the TNR is lower. This

Input parameter	Value
Number of channels	3
Input window	0.5s
Input step	0.25s
Number of layers	3
Number of neurons per layer	120, 60, 30
Pre-ictal time	5s
Number of epochs	76
Hidden layer activation function	LReLu ($\alpha = 0.2$)
Output layer activation function	Softmax
Loss function	Categorical loss
Output metric	**Value**
Average Prediction Rate	0.65
True Positive Rate	0.92
True Negative Rate	0.38
Prediction time	6.1s
Categorical cross-entropy loss	0.50

Table 5.5: Metrics of best run using 6 features for each channel as input

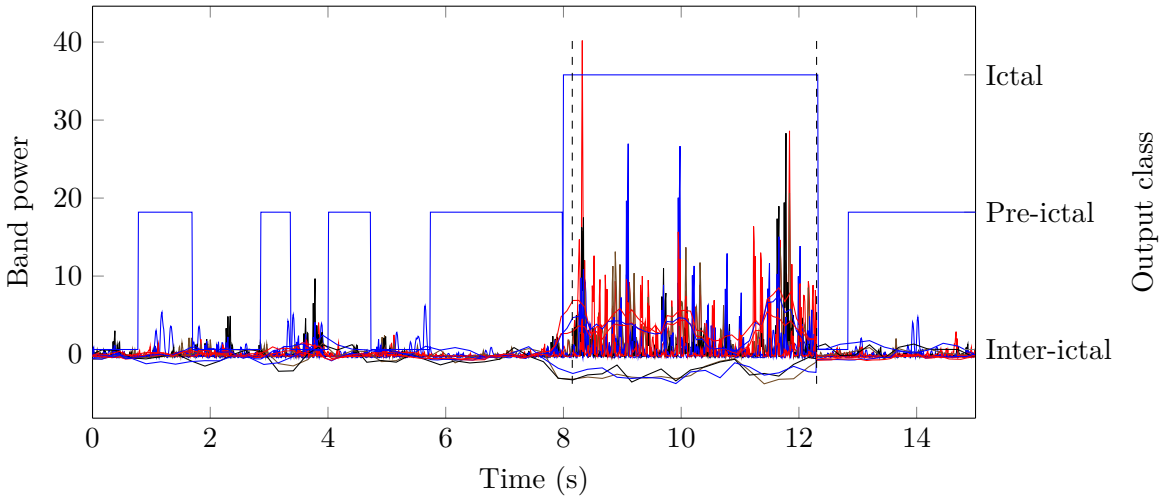


Figure 5.18: A sample of the predicted class output

means that more seizures are being detected using this input set, while also having more false-positives.

5.4.2 Time-To-Event exploratory results

This section describes the results of an experiment using a different way of predicting seizures. The output of the neural network is a Time-To-Event (TTE): a prediction

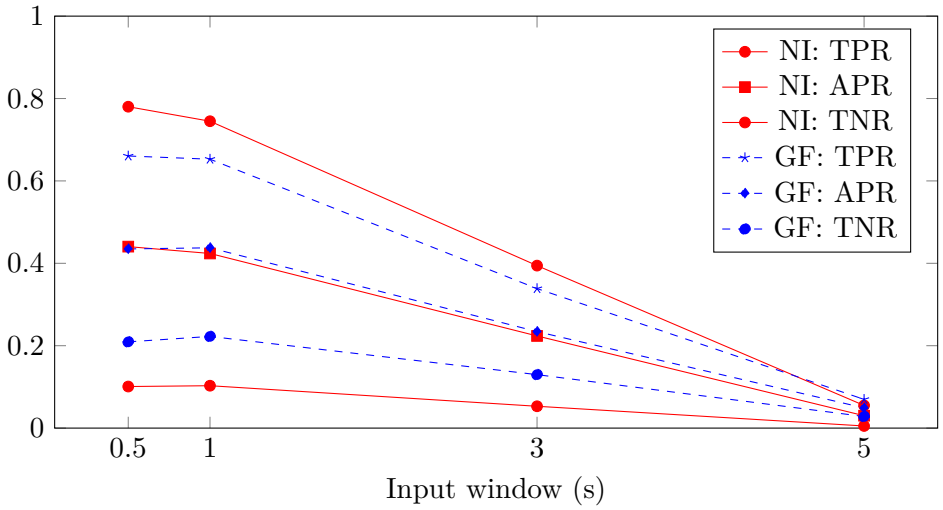


Figure 5.19: Resulting Avg. APR/TNR/TPR for multiple input windows and input data sets. (Red lines = Normalised input, blue lines = generated features)

of the time until the next event, which in a perfect situation should gradually decrease until the event is encountered. This time is depicted by two parameters of the Weibull distribution, and special activation and loss functions are used in this approach. See subsection 2.5.6 for more information on the Weibull Time-To-Event Recurrent Neural Network (WTTE-RNN) type of network.

The amount of data used for this experiment is less than for the previous set of experiments. The reason behind this is that the model is not behaving totally stable, resulting in infinite losses or losses that are not a number (NaN). This was not solved by using different sub-sets of the available input-data, which makes it unlikely that an artifact in the input data is causing the instability. Also other parameters like the learning rate did not influence the stability of the model. As losses are the main metric used to train a model, it was not possible to train any further or predict any output with a non-converging model. Only limiting the amount of input-data and number of epochs resulted in a converging model, therefore we limited the amount of data to 208 minutes of train data and 5 minutes of test data. With this amount of data is was possible to train for 75 epochs while the neural network converged. The input parameters are listed in Table 5.6.

The results of this experiment are exploratory, meaning that we only ran the model with a single input set and single set of parameters, to explore it's possibilities. We did also not use a grid-search to get to the best combination of parameters possible, nor automatic metric calculation to determine the performance of the network. The goal of this experiment is to see whether it's feasible to predict a Time-To-Event based on an input of normalised ECoG recordings. A sample of the input and output is plotted in Figure 5.20, where we see the input, expected output (blue line) and predicted output (red line). A sharp drop can be seen in the predicted output a few seconds before the actual seizure, which indicates that seizures can possibly be predicted using

Parameter	Values
Network type	GRU
Input type	3 channels normalised data
Output	Weibull distribution parameters λ and k
Number of channels	3
Input window	0.5s
Input step	0.033s
Number of layers	1
Number of neurons per layer	5
Pre-ictal time	4s
Number of epochs	75
Hidden layer activation function	Tan-h
Output layer activation function	Exponential (α neuron) and soft-plus (β neuron)
Loss function	Log-likelihood

Table 5.6: Overview of the TTE model input parameters

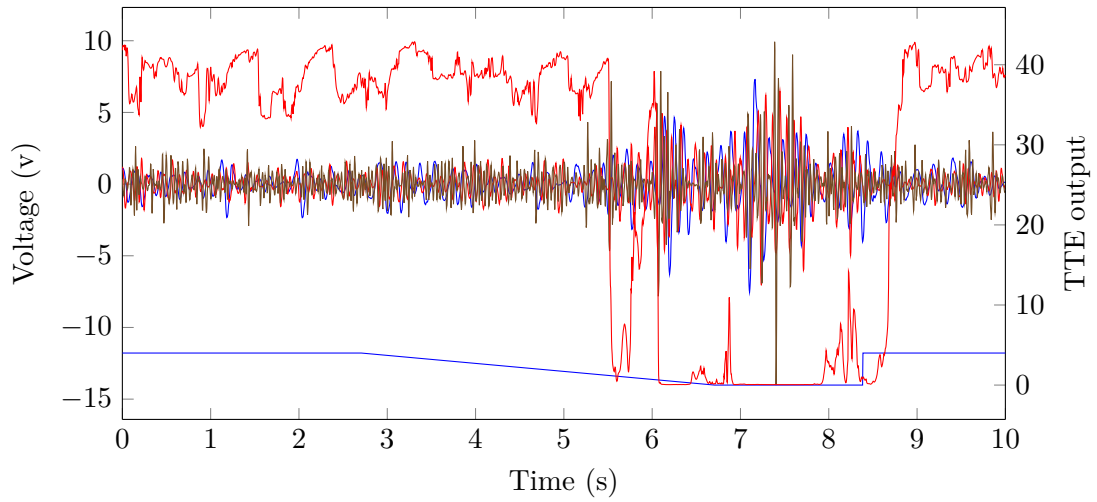


Figure 5.20: A sample of the predicted TTE output

a WTTE-RNN.

5.5 Discussion

In this section we will discuss the results as presented in the previous section.

The experiments had as goal to see if it is possible to predict absence seizures using a neural network. A grid-search has been performed to try out all different combinations of input parameters. The results are summarised using four metrics, the APR, TNR, TPR, the detection delay and the network categorical cross-entropy loss.

Looking at these results we can identify several trends.

1. The network size and network type do not have a large influence on the resulting APR. This can be seen in the graphs when plotted against the input window: all lines have the same trend and lie close to each other.

2. The prediction performance decreases with an increasing input window. This effect is seen across all experiments and is visible in all graphs with the input window on the x-axis.

3. The TPR is a lot higher than the TNR. This is true for both data sets across all experiments.

4. The categorical cross-entropy loss is the lowest for the largest network sizes, with a first-layer size of 120 and input window of 5s. The loss becomes higher when decreasing the layer size or input window. This is visible in the loss-graphs for all experiments.

5. The best performance in terms of prediction time is when using an input window of 3 seconds.

The first two findings are contrary to the expectations. With an increasing network size or input size, one expects better results up to a certain optimum, since there is more information available to base the prediction on. With only a small data-set, the network is prone to over-fitting as it can not learn a more generalised model based on a wide range of input cases. When increasing the amount of available information in the network too much, the network will become unable to relate this information to each other and the performance will drop. In between these two situations we expect an optimum. But in our findings the optimum amount of information is the smallest amount tried, which can indicate that the network cannot relate information at all.

The third finding acknowledges this. The TNR is very low across all experiments, meaning that there are a lot of false-positive seizure predictions in the network output. During the inter-ictal period, the network is unable to relate input data with a certain output class, resulting in random inter-ictal and pre-ictal period predictions. In other words, there seems to be nothing in the data that distinguishes between an inter-ictal or pre-ictal period. The ictal period, the seizure itself, is detected quite decently, but it was already known that seizures could be detected using neural networks. This however leads to the relatively high TPR.

The fourth finding shows how the network predicts based on learned behaviour or random guesses. A high loss means that the predicted output class is not the same as the annotated output class for a lot of samples, while a lower loss means that it is correct for more samples. A high loss indicates that the network did not learn well enough from training and is effectively randomly guessing the output class. A lower loss for larger network sizes means that during training the network was better able to learn the relation between input and output. As shown in the graphs above, the APR decreases when the loss decreases, which means that the network was unable to relate input and output data in a proper way. For a larger network, the amount of data and neurons in the network is higher, which will lead to a more stable output because the prediction is based on more data. When in that case the APR is still low with a low cross-entropy

loss, the network either over-fitted or was unable to learn the right relation between input and output data.

In the last finding, we do see the effect of an optimum. The prediction time is the highest when using an input window of 3 seconds, while larger or smaller input windows yield a lower prediction time. Yet, based on the very poor performance of detecting pre-ictal periods, the pre-ictal detections which have led to a larger prediction time, can as well be false-positives. These false-positives precede a ictal-period and their durations are therefore added to the prediction time of the particular seizure.

The experiments in this work are limited by the used data recordings and computational power. The data used contains only 3-channels ECoG measurements of three locations in the brain. Having more channel data might have given more insight into the propagation of the seizure onset through the brain. Also the recording are done on mutant mice with very periodic seizures, about one seizure every minute. Data with high periodicity prevents the use of large input windows and can mean that the previous seizure still influences the inter-ictal period. This makes it harder to differentiate between the inter-ictal and pre-ictal period. For a grid-search approach a lot of computational power is needed as the number of combinations of input parameters easily becomes very large. We had to limit the amount of combinations to keep evaluating them feasible.

5.5.1 WTTE-RNN results discussion

Only exploratory research has been done on TTE prediction using a WTTE-RNN, making for a small result set. No actual metrics have been evaluated on the results, because the test-data contain only a few seizures. An example of the output data has been given in Figure 5.20. An indication of a precursor can be seen in the sudden drop of the predicted TTE. This shows that it is possible to predict a seizure on a small data-set, but it is not known how generic the model is and whether it only works on this subset of the data.

6 Conclusions

This work is mainly focused on applying neural networks to Electrocorticography (ECoG) data in order to predict absence seizures. The grid-search approach has resulted in multiple experiments with their results, which are evaluated in the previous chapter. In this chapter we conclude the contributions of this thesis.

6.1 Thesis overview

The goal of this thesis has been to evaluate if it is possible to predict absence seizures based on 3-channel ECoG measurements using several neural-network algorithms. Specifically, to determine the best-performing set of input data, neural-network type and parameters for predicting absence seizures. To be able to do this, the experiments were split up in two different parts: classification of inter-ictal, pre-ictal and ictal periods and Time-To-Event (TTE) prediction using the Weibull distribution.

The classification experiments were split again into two parts, both covering a certain set of input features, generated from the raw ECoG recordings. The first part used normalised recordings as input, while the second part used 3 different features generated from the input data: the power of Discrete-Wavelet Transform (DWT) bands, the Approximate Entropy (ApEn) and the variance of the recordings. The classification was done with three different types of neural networks: the Multi-Layer Perceptron (MLP) network, the Long Short-Term Memory (LSTM) network and the Gated Recurrent Unit (GRU) network. For each network all combinations of multiple input parameters were tried to find an optimal functioning neural network. Performance was evaluated in terms of Average Prediction Rate (APR), True Positive Rate (TPR), True Negative Rate (TNR), prediction time and the loss of the neural network during training.

When looking at the results of the experiments with normalised input, there was no clear winner in terms of performance. Also, the best-performing combination in terms of APR and prediction time did not perform well enough to be used in practice. Using normalised recordings as input, the best-performing combination of parameters yielded an APR of 0.57 and a prediction time of 3.1 seconds. Detection did work well as the TPR was 1.

Using the generated features as input, the results were slightly better. The resulting APR was 0.65 and the prediction time 6.1 seconds. The prediction time however could be due to a lot of false positive inter-ictal predictions, as the TNR was only 0.38. Detection did work well as the TPR was 0.92.

Based on the results, we have to conclude that by using the approach set-out in this thesis with the used data recordings, it was not possible to predict upcoming absence seizures in a reliable way. However, the results using a Weibull Time-To-Event Recurrent Neural Network (WTTE-RNN) look promising, but need further research.

6.2 Contributions

The results of this thesis have its limitations in terms of accuracy and prediction time, and do not make for a usable system using the type of measurements and neural-network parameters used in this work. The machine-learning algorithms were not able to distinguish clearly enough between the inter-ictal, pre-ictal and ictal periods. Especially deviating between the inter-ictal and pre-ictal periods turned out to be difficult, because their characteristics lie close to each other. However, some important contributions have been made by this thesis:

- Recurrent Neural Networks (RNNs) were applied on ECoG recordings, using a grid-search approach to compare multiple combinations of neural-network parameters, in order to predict absence seizures. Previous research on using machine learning and neural networks specifically, used mainly Support Vector Machines (SVMs) and MLP networks for predicting upcoming seizures. This research however, applied RNNs to be able to incorporate time-domain data in a better way. Although the trained networks in this work are not reliable enough to proof RNNs can be used for seizure predictions, they show some results on which future works can be based.

- Research has been done on which features can be generated from ECoG data, to be able to show distinctive characteristics of ECoG measurements.

- A guideline is provided for future research on machine-learning based seizure-prevention approaches, by setting up an extensive experimental setup and by providing a thorough analysis of working approaches and limitations.

- A regression method was used to predict a TTE value depicting the time to the next seizure, using Weibull distribution parameters, in combination with a RNN. The combination of using an RNN together with the regression of distribution parameters for prediction of events is not applied before on predicting absence seizures based on ECoG data.

6.3 Future work

The work presented in this thesis can be considered as another step taken into the direction of reliable absence seizure prediction. Although the results of this thesis were not positively on itself, they allow for several recommendations for future work.

- The recordings can be combined with sensory recordings. Now only 3-channel ECoG data is used which can also be extended with more channels of ECoG recordings and other recordings like Electromyography (EMG) and Electroencephalography (EEG). EMG will give insight into the muscle activity of the subject, and can also show if spasms or lack of muscle activity occur before or during a seizure. EEG recordings give insight into the heart rhythm, of which variations can be related to an upcoming seizure.

- When more channels of ECoG data are available, it is also possible to take into account the recording locations. This spatial information can be used to relate seizure precursors not only to time and waveform, but also to the location in the brain. This might give insights into the propagation of the possible precursor and seizure itself through the brain, and possibly lead to better prediction performance.
- Another recommendation is to use data with less frequent seizures. When seizures are occurring every minute, there is not a lot of data providing samples for learning to classify the inter-ictal period. Especially as we do not know exactly the size of the pre-ictal window. Having more data in between seizures gives more room to vary the prediction time and pre-ictal time window.
- The exploratory WTTE-RNN experiment showed a promising result, but further investigation is needed to see if it is actually a feasible method. The model needs to be able to converge on larger data-sets, which in turn can prevent over-fitting. Also, the described evaluation method needs to be tested to see if it is a method which is usable in practise. When these problems are overcome, it is possible to evaluate the performance of the WTTE-RNN and conclude whether it is usable at all for seizure prediction.

Bibliography

[1] R. S. Fisher, W. Van Emde Boas, W. Blume, C. Elger, P. Genton, P. Lee, and J. Engel, "Epileptic seizures and epilepsy: Definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE)," *Epilepsia*, vol. 46, no. 4, pp. 470–472, 2005.

[2] C. P. Panayiotopoulos, "Typical absence seizures and their treatment," *Archives of Disease in Childhood*, vol. 81, no. 4, pp. 351–355, 1999.

[3] J. R. Tenney and T. A. Glauser, "The current state of absence epilepsy: Can we have your attention?" *Epilepsy Currents*, vol. 13, no. 3, pp. 135–140, 2013.

[4] L. Kros, O. H. Eelkman Rooda, J. K. Spanke, P. Alva, M. N. Van Dongen, A. Karapatis, E. A. Tolner, C. Strydis, N. Davey, B. H. Winkelman, M. Negrello, W. A. Serdijn, V. Steuber, A. M. Van Den Maagdenberg, C. I. De Zeeuw, and F. E. Hoebeek, "Cerebellar output controls generalized spike-and-wave discharge occurrence," *Annals of Neurology*, vol. 77, no. 6, pp. 1027–1049, 2015.

[5] C. P. Panayiotopoulos, "Typical absence seizures and related epileptic syndromes: assessment of current state and directions for future research." *Epilepsia*, pp. 2131–2139, 2008. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1111/j.1528-1167.2008.01777.x/full

[6] G. Giannakakis, V. Sakkalis, M. Pediaditis, C. Farmaki, P. Vorgia, and M. Tsiknakis, "An approach to absence epileptic seizures detection using Approximate Entropy," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, vol. 2013. IEEE, jul 2013, pp. 413–416. [Online]. Available: http://ieeexplore.ieee.org/document/6609524/

[7] S. M. Pincus, "Approximate entropy as a measure of system complexity," *Mathematics*, vol. 88, no. March, pp. 2297–2301, 1991.

[8] Y. H. Pan, Y. H. Wang, S. F. Liang, and K. T. Lee, "Fast computation of sample entropy and approximate entropy in biomedicine," *Computer Methods and Programs in Biomedicine*, vol. 104, no. 3, pp. 382–396, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.cmpb.2010.12.003

[9] M. S. Raza and U. Qamar, *Understanding and Using Rough Set Based Feature Selection: Concepts, Techniques and Applications*. Springer, 2017.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, sep 1995. [Online]. Available: http://link.springer.com/10.1007/BF00994018

[11] P. Cao, D. Zhao, and O. Zaiane, "An optimized cost-sensitive SVM for imbalanced data learning," *Lecture Notes in Computer Science (including subseries Lecture*

Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7819 LNAI, no. PART 2, pp. 280–292, 2013.

[12] K.-B. Duan and S. S. Keerthi, "Which Is the Best Multiclass SVM Method? An Empirical Study," 2005, pp. 278–285. [Online]. Available: http://link.springer.com/10.1007/11494683{_}28

[13] U. Orhan, M. Hekim, and M. Ozer, "EEG signals classification using the K-means clustering and a multilayer perceptron neural network model," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 475–13 481, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2011.04.149

[14] M. Jordan, "Serial order: A parallel distributed processing approach," *Advances in Psychology*, vol. 121, no. 667, pp. 471–495, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166411597801112

[15] J. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 1 990, pp. 179–211, 1990. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/036402139090002E{%}0Apapers2://publication/uuid/BF1D3C8F-9A67-4350-ACE9-30BDA3C1FEC8

[16] D. Zipser and R. J. Williams, "Gradient-Based Learning Algorithms for Recurrent Networks and Thei r Computational Complexity," *Back-propagation: Theory, Architectures and Applications*, pp. 433–486, 1995.

[17] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," pp. 157–166, 1994.

[18] S. Hochreiter and J. Urgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: http://www7.informatik.tu-muenchen.de/{~}hochreit{%}5Cnhttp://www.idsia.ch/{~}juergen

[19] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[20] E. Martinsson, "WTTE-RNN : Weibull Time To Event Recurrent Neural Network," *University of Gothenburg*, no. Department of Computer Science and Engineering, 2016.

[21] W. Weibull, "A statistical distribution function of wide applicability," *Journal of applied mechanics*, vol. 18, no. 4, pp. 293–297, 1951. [Online]. Available: http://web.cecs.pdx.edu/{~}cgshirl/Documents/Weibull-ASME-Paper-1951.pdf

[22] Y. Zhou, R. Huang, Z. Chen, X. Chang, J. Chen, and L. Xie, "Application of approximate entropy on dynamic characteristics of epileptic absence seizure," *Neural Regeneration Research*, vol. 7, no. 8, pp. 572–577, mar 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4346979/

[23] H. Ocak, "Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy," *Expert Systems with Applications*, vol. 36, no. 2 PART 1, pp. 2027–2036, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2007.12.065

[24] S. F. Liang, W. L. Chang, and H. Chiueh, "EEG-based absence seizure detection methods," *Proceedings of the International Joint Conference on Neural Networks*, 2010.

[25] V. Sakkalis, G. Giannakakis, C. Farmaki, A. Mousas, M. Pediaditis, P. Vorgia, and M. Tsiknakis, "Absence seizure epilepsy detection using linear and nonlinear eeg analysis methods," *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*, pp. 6333–6336, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6611002

[26] E. B. Petersen, J. Duun-Henriksen, A. Mazzaretto, T. W. Kjar, C. E. Thomsen, and H. B. Sorensen, "Generic single-channel detection of absence seizures," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 4820–4823, 2011.

[27] A. Subasi, "Epileptic seizure detection using dynamic wavelet network," *Expert Systems with Applications*, vol. 29, no. 2, pp. 343–355, 2005.

[28] ——, "Application of adaptive neuro-fuzzy inference system for epileptic seizure detection using wavelet feature extraction," *Computers in Biology and Medicine*, vol. 37, no. 2, pp. 227–244, 2007.

[29] K. Zeng, J. Yan, Y. Wang, A. Sik, G. Ouyang, and X. Li, "Automatic detection of absence seizures with compressive sensing EEG," *Neurocomputing*, vol. 171, pp. 497–502, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2015.06.076

[30] 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.

[31] L. Diambra and C. P. Malta, "Nonlinear models for detecting epileptic spikes," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 59, no. 1, pp. 929–937, 1999.

[32] J. M. Sorokin, J. T. Paz, and J. R. Huguenard, "Absence seizure susceptibility correlates with pre-ictal β oscillations," *Journal of Physiology Paris*, vol. 110, no. 4, pp. 372–381, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jphysparis.2017.05.004

[33] F. Mormann, R. G. Andrzejak, C. E. Elger, and K. Lehnertz, "Seizure prediction: The long and winding road," *Brain*, vol. 130, no. 2, pp. 314–333, 2007.

[34] T. Netoff, Y. Park, and K. Parhi, "Seizure prediction using cost-sensitive support vector machine." *Conference proceedings : ... Annual International Conference*

of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference, vol. 2009, pp. 3322–5, 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19964303

[35] "EEG Database at Epilepsy Center of the University Hospital of Freiburg, Germany," 2003. [Online]. Available: http://epilepsy.uni-freiburg.de/freiburg-seizure-prediction-project/eeg-database

[36] L. Chisci, A. Mavino, G. Perferi, M. Sciandrone, C. Anile, G. Colicchio, and F. Fuggetta, "Real-time epileptic seizure prediction using AR models and support vector machines," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 5, pp. 1124–1132, 2010.

[37] Y. Park, L. Luo, K. K. Parhi, and T. Netoff, "Seizure prediction with spectral power of EEG using cost-sensitive support vector machines," *Epilepsia*, vol. 52, no. 10, pp. 1761–1770, 2011.

[38] K. Gadhoumi, J.-M. Lina, and J. Gotman, "Seizure prediction in patients with mesial temporal lobe epilepsy using EEG measures of state similarity," *Clinical Neurophysiology*, vol. 124, no. 9, pp. 1745–1754, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.clinph.2013.04.006{%}5Cnhttp://linkinghub.elsevier.com/retrieve/pii/S1388245713002733

[39] J. J. Howbert, E. E. Patterson, S. M. Stead, B. Brinkmann, V. Vasoli, D. Crepeau, C. H. Vite, B. Sturges, V. Ruedebusch, J. Mavoori, K. Leyde, W. D. Sheffield, B. Litt, and G. A. Worrell, "Forecasting Seizures in Dogs with Naturally Occurring Epilepsy," *PLoS ONE*, vol. 9, no. 1, p. e81920, jan 2014. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0081920

[40] C. Alexandre Teixeira, B. Direito, M. Bandarabadi, M. Le Van Quyen, M. Valderrama, B. Schelter, A. Schulze-Bonhage, V. Navarro, F. Sales, and A. Dourado, "Epileptic seizure predictors based on computational intelligence techniques: A comparative study with 278 patients," *Computer Methods and Programs in Biomedicine*, vol. 114, no. 3, pp. 324–336, may 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cmpb.2014.02.007http://linkinghub.elsevier.com/retrieve/pii/S0169260714000571

[41] N. Moghim and D. W. Corne, "Predicting Epileptic Seizures in Advance," *PLoS ONE*, vol. 9, no. 6, p. e99334, jun 2014. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0099334

[42] C. Alvarado-Rojas, M. Valderrama, A. Fouad-Ahmed, H. Feldwisch-Drentrup, M. Ihle, C. A. Teixeira, F. Sales, A. Schulze-Bonhage, C. Adam, A. Dourado, S. Charpier, V. Navarro, and M. Le Van Quyen, "Slow modulations of high-frequency activity (40–140 Hz) discriminate preictal changes in human focal epilepsy," *Scientific Reports*, vol. 4, no. 1, p. 4545, may 2015. [Online]. Available: http://www.nature.com/articles/srep04545

[43] G. Ouyang, X. Li, C. Dang, and D. A. Richards, "Using recurrence plot for determinism analysis of EEG recordings in genetic absence epilepsy rats," *Clinical Neurophysiology*, vol. 119, no. 8, pp. 1747–1755, 2008.

[44] A. Lüttjohann, A. Lüttjohann, V. V. Makarov, V. A. Maksimenko, A. A. Koronovskii, and A. E. Hramov, "Methods of automated absence seizure detection, interference by stimulation, and possibilities for prediction in genetic absence models," *Journal of Neuroscience Methods*, vol. 260, pp. 144–158, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jneumeth.2015.07.010

[45] L. D. Iasemidis, S. Deng-Shan, W. Chaovalitwongse, J. C. Sackellares, P. M. Pardalos, J. C. Principe, P. R. Carney, A. Prasad, B. Veeramani, and K. Tsakalis, "Adaptive epileptic seizure prediction system," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 5, pp. 616–627, 2003. [Online]. Available: http://ieeexplore.ieee.org/ielx5/10/26967/01198251.pdf?tp={&}arnumber=1198251{&}isnumber=26967

[46] M. D'Alessandro, R. Esteller, G. Vachtsevanos, A. Hinson, J. Echauz, and B. Litt, "Epileptic seizure prediction using hybrid feature selection over multiple intracranial EEG electrode contacts: a report of four patients." *IEEE transactions on bio-medical engineering*, vol. 50, no. 5, pp. 603–615, 2003.

[47] B. Litt, R. Esteller, J. Echauz, M. D'Alessandro, R. Shor, T. Henry, P. Pennell, C. Epstein, R. Bakay, M. Dichter *et al.*, "Epileptic seizures may begin hours in advance of clinical onset: a report of five patients," *Neuron*, vol. 30, no. 1, pp. 51–64, 2001.

[48] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou, "On the Class Imbalance Problem," *International Conference on Natural Computation*, pp. 192–201, 2008.

[49] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical Review A*, vol. 45, no. 6, pp. 3403–3411, mar 1992. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.39.5121https://link.aps.org/doi/10.1103/PhysRevA.45.3403

[50] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," 2015. [Online]. Available: http://arxiv.org/abs/1505.00853

[51] F. Chollet, "Keras: Deep learning library for theano and tensorflow," 2015.

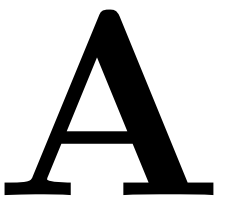
[52] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[53] "Molecular devices," https://www.moleculardevices.com/, accessed: 2019-12-28.

[54] S. Garcia, D. Guarino, F. Jaillet, T. R. Jennings, R. Pröpper, P. L. Rautenberg, C. Rodgers, A. Sobolev, T. Wachtler, P. Yger *et al.*, "Neo: an object model for handling electrophysiology data in multiple formats," *Frontiers in neuroinformatics*, vol. 8, p. 10, 2014.

[55] F. Mormann, R. G. Andrzejak, T. Kreuz, C. Rieke, P. David, C. E. Elger, and K. Lehnertz, "Automated detection of a preseizure state based on a decrease in synchronization in intracranial electroencephalogram recordings from epilepsy patients," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 67, no. 2, p. 10, 2003.

Appendix: WBAN survey



This survey has been originally commissioned to explore the current state of modern Wireless Body Area Networks (WBANs). The use of multimodal data, recorded by sensors on other body-parts, which are being communicated over a WBAN, would possibly add to the accuracy of seizure prediction. However, our expectations fell short when we realized that only single-mode, ECoG data was available to us for using in this thesis. Still, we chose to append this WBAN survey as part of this thesis for the following reasons:

a It was work also done during the thesis period, and

b It gives hints on the level of technological maturity of modern WBANs

Hopefully WBANs can be used in future seizure-prediction systems.

Wireless Body Area Networks: state of the art

Corniël Joosse

April 25, 2017

1 Introduction

A Wireless Body Area Network (WBAN) is a network of sensor and actuator nodes which are wireless connected, and are deployed in the vicinity of, or inside the body. Such a network can be used to monitor all sorts of biomedical signals, e.g. heart rate, respiration rate, muscle activity or leg movement. Certain analyses can be done by combining the data from different nodes, either remotely or within the network, providing direct feedback to other nodes in the network. One has to solve multiple challenges before a WBAN will work properly. Battery life, quality of service, availability en security among other things have to be guaranteed. In this research we give an overview of the current state of WBANs.

The first publication about WBANs dates from 2002 and identifies the main difference between the existing medical wireless systems that use point-to-point communication and a WBAN with multiple nodes in a network [1]. It also mentiones that previous research at that time has only been done in the field of Personal Area Networks (PANs), which are more focused on the personal environment and not the body itself. The following years there have been numerous publications about WBAN systems, methods and possible protocols.

When we look at the trends based on the number of publications about WBANs, which can be found on the website of Web of Science[1], we see that the research interest in WBANs rises. Compared to Wireless Sensor Networks (WSNs) and medical implants (IMDs) there are still a lot less publications (see Figure 1a), but the amount of publication is increasing steadily, and even increases faster for WBANs than for WSNs and IMDs as can be seen in Figure 1b. There is a large dip in 2010, but the reason for this is not really clear. It can be due to the economic crisis which started around 2007. Effects of a lack of research funds can be seen after two to three years.

The percentage of the total publications published per year is plotted in Figure 1c, this is the same data as depicted in Figure 1a but then normalised to a cumulative total of 100%. We can see that more than 20 percent of the publications on WBANs is published in 2015. We do not plot the data of 2016 because the index can still be incomplete due to the delays between submission, publication and indexation of a paper. If we bin the number of publications on WBANs per two years (Figure 2), a stable increase can be seen with up to 932 publications in 2014 and 2015. We choose a binning period of two years because the time between research and publication is typically two years.

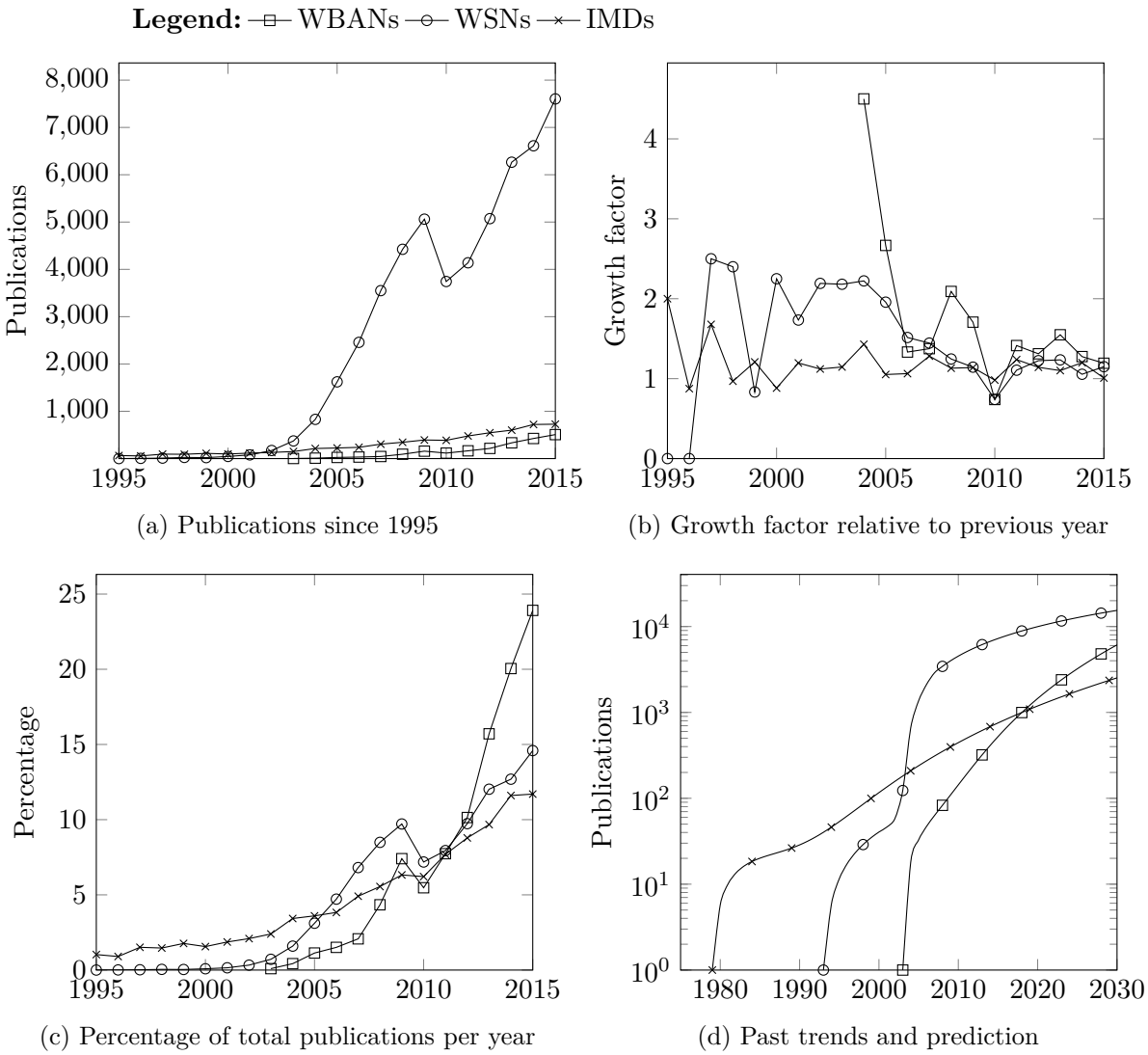
The trends are based on data from Web of Science. The data can be exported from their website, and the key to the right data is to use the right search terms. We have used the following search queries to get to the plotted trends:

- **WBANs**: `` `wireless "body area" network* OR WBAN` `` (988 results)

[1] http://apps.webofknowledge.com

- **WSNs**: `'wireless sensor network*'` (22091)
- **IMDs**: `'medical implant* device*'` (6026 results)

The asterisks are used to match zero or multiple characters, to match multiple versions of a word, e.g. both singular and plural terms.



(a) Publications since 1995

(b) Growth factor relative to previous year

(c) Percentage of total publications per year

(d) Past trends and prediction

Figure 1: Publication statistics (from Web of Science)

The total number of publications reported by Web of Science on Wireless Body Area Networks is 2120, which is not even 4% of the publications on WSNs. This is mainly due to being a newer subject, but also because WSNs is a broader topic and gets more interest in general.

In Figure 1d we have plotted a trend-line for each of the three subjects, and extrapolated it until 2030. The trend-lines are formed by fitting a second or third order polynomial on the current data. For the first 10 years of the trend-line for IMDs, a linear trend-line is used, because the research interest did not follow a polynomial form in that period. The y-axis is scaled logarithmic because the trends are more or less logarithmic, so the data is better visible this way. Note here that this is a very rough estimation, and does not have to reflect the future in any way.

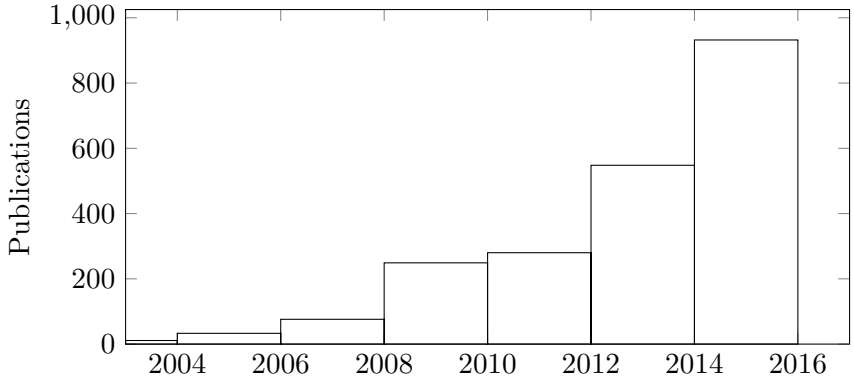


Figure 2: Number of WBAN publications binned per 2 years (Web of Science)

The trend-lines show us that nowadays Wireless Sensor Networks is the most popular topic, and if the research interest will follow the current trend, it will also be like that in the future. Although the first publications on Implantable Medical Devices date from further back, WSNs overtake the lead in 2004.

In the next chapters we will show an overview of the surveyed systems, the results from the publications about the number of nodes, used topologies, wireless communication choices, routing protocols, security and conclude with a short discussion.

2 Overview of surveyed systems

To find usable publications about complete WBAN systems we have mainly used Google Scholar[2] and the digital library from IEEE Xplore[3]. Filtering of the results is done manually by looking at the title: if the title implies a subsystem suitable for or in a WBAN, than it is not a complete system. But if it states an application or use case for a WBAN or using a WBAN, than it possibly is a complete system. Filtering is done manually this way, and the only used search terms are '`Body Area Network*`' and '`Wireless Body Area Network*`'.

All 53 surveyed systems are listed in Table 1. For every system an application or design focus is listed. Also the approach on how the researchers came to their design is interesting, because it influences the design choices of the WBAN. We can group the publications in the following categories based on their approach:

- **WBAN architecture focus**: The publication describes a top-down approach, starting with the architecture of the Body Area Network. These publications generally describe a general purpose system or a patient monitoring system.
- **Global architecture focus**: This is also a top-down approach, but the publication also includes the global architecture including beyond-WBAN communication and cloud computing. The applications for such approaches are mainly remote patient or elderly monitoring.
- **Application driven**: The publications in the application driven category use a bottom-up approach, starting with the application. All design choices and development efforts are based on the application.

[2]https://scholar.google.com
[3]http://ieeexplore.ieee.org

- **Specific component focus**: These publications highlight a certain component of WBANs and focus on the design and development of that particular component. The general WBAN architecture is only used to utilise the component. This is also a bottom-up approach, but then to proof a certain concept.

Figure 3 shows the distribution of the different categories, and as we can see it is quite even distributed. There are a bit more publications which focus on the WBAN architecture, and just a few publications about specific components.

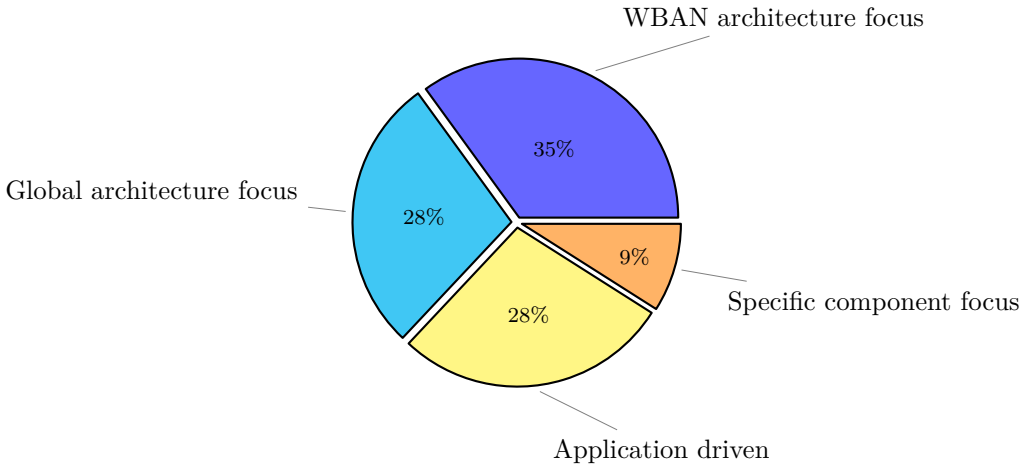


Figure 3: Application and design focus categories of WBAN systems (sample size: 53)

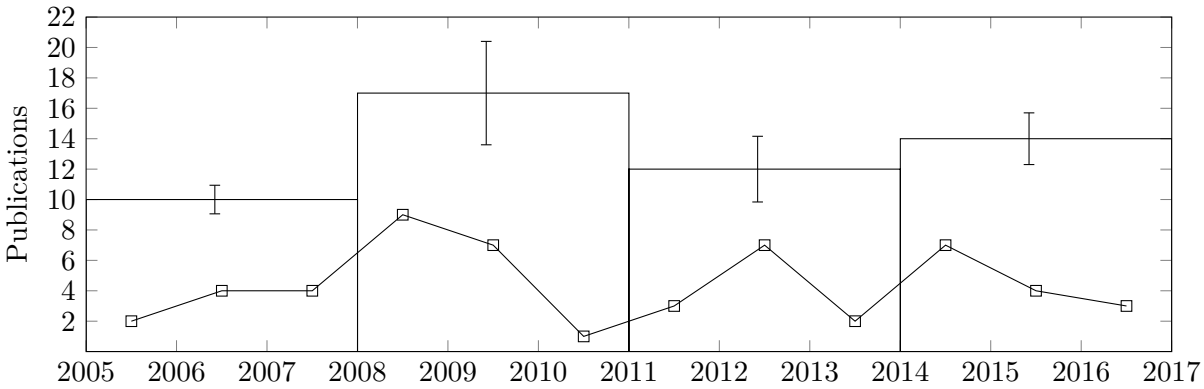


Figure 4: Surveyed publications per year (53 in total)

The number of surveyed publications per year is plotted in Figure 4. Although the sample size is just 53 publications, we can see that more papers describing complete systems were published between 2008 and 2011. Later publications might focus more on specific subsystems of a WBAN, e.g. a dedicated low-power microcontroller or security aspects without applying it to a real WBAN system.

Application/focus	**Authors**		**Year**
General medical WBAN	V. Shnayder, B. Chen et al.	[2]	2005
Assisted rehabilitation	E. Jovanov, A. Milenkovic et al.	[3]	2005
Remote monitoring	T. Falck, J. Espina et al.	[4]	2006
Using MEMS technology	F.E.H. Tay, M. N. Nyan et al.	[5]	2006

Low power WBAN	B. Gyselinckx, R. Vullers et al.	[6]	2006
Ambulatory monitoring	C. Otto, A. Milenkovic et al.	[7]	2006
Triage system	T. Gao, T. Massey et al.	[8]	2007
Elderly monitoring	S. Saadaoui, L. Wolf	[9]	2007
Performance analysis	D. Domenicali, M.G. Di Benedetto	[10]	2007
Sleep staging	N. de Vicq, F. Robert	[11]	2007
Patient monitoring	E. Montón, J.F. Hernandez et al.	[12]	2008
Hip surgery rehabilitation	M. Soini, J. Nummela et al.	[13]	2008
Remote monitoring	L. Xuemei, J. Liangzhong et al.	[14]	2008
Telerehabilitation	M. Hamel, R. Fontaine et al.	[15]	2008
Activity recognition	E. Farella, A. Pieracci et al.	[16]	2008
Two-tier WBAN	S. Jiang, Y. Cao et al.	[17]	2008
Performance analysis	M. Sukor, S. Ariffin et al.	[18]	2008
Remote monitoring	R.A. Rashid, S.H.S. Arifin et al.	[19]	2008
Patient monitoring	J.Y. Khan, M. R. Yuce et al.	[20]	2008
UWB architecture	R. Chávez-Santiago, A. Khaleghi et al.	[21]	2009
General WBAN	B. Wang, L. Wang et al.	[22]	2009
Patient monitoring	A. Saeed, M. Faezipour et al.	[23]	2009
Nervous system monitoring	L. Brown, B. Grundlehner et al.	[24]	2009
Sensor node development	A.T. Barth, M.A. Hanson et al.	[25]	2009
Elderly fall assesment	T. O'Donovan, J. O'Donoghue et al.	[26]	2009
Customizable WBAN	K. Wac, R. Bults et al.	[27]	2009
Patient monitoring	M. Yuce,	[28]	2010
Remote monitoring	E. Katoch, M. Smole et al.	[29]	2011
Remote monitoring	S. Sharma, A.L. Vyas et al.	[30]	2011
Cycling monitoring	R. Marin-Perianu, M. Marin-Perianu et al.	[31]	2011
Sleep disorders detection	A. Nassir, O. Barnea	[32]	2012
Smartphone based WBAN	M. Wagner, B. Kuch et al.	[33]	2012
Remote monitoring	C. Wang, Q. Wang et al.	[34]	2012
General WBAN	U. Mitra, B.A. Emken et al.	[35]	2012
Highly reliable WBAN	Y. Hamada, K. Takizawa et al.	[36]	2012
Pulse wave velocity tracking	K. Li, S. Warren	[37]	2012
General monitoring	S.L. Tan, J. García-Guzmán et al.	[38]	2012
Remote monitoring	P. Dinkar, A. Gulavani et al.	[39]	2013
Monitoring fitness exercises	Y. Varatharajah, N. Karunathilaka et al.	[40]	2013
Smarphone based WBAN	Y. Shi, Y. Zhang	[41]	2014
General WBAN	M. Chen, Z. Li et al.	[42]	2014
Activity recognition	Z. He, X. Bai	[43]	2014
Patient monitoring	U. Ghoshdastider, R. Viga et al.	[44]	2014
Dual-band WBAN	K.M.S. Thotahewa, J.M. Redouté et al.	[45]	2014
Patient monitoring	B.R. Nandkishor, A. Shinde et al.	[46]	2014
Monitoring Parkinsons Disease	Z. Dong, H. Gu et al.	[47]	2015
Aeronautical	T. Przybylski, P. Froehle et al.	[48]	2015
Remote monitoring	J.A. Hidalgo, A. Cajiao et al.	[49]	2015
Heart attack detection	G. Wolgast, C. Ehrenborg et al.	[50]	2016
Swimming monitoring	R. Li, Z. Cai et al.	[51]	2016
Real-time WBAN	Y. Wang, Y. Zheng et al.	[52]	2016

Table 1: All surveyed systems with application/focus specified

3 Number of nodes

The number of nodes used in the surveyed systems varies from 1 to 31 with an average of 4.92 nodes per system including the master node. To see whether there is a relation between the number of nodes and the publication year, we have plotted the average number of nodes for each year, see Figure 5. The error bars indicate the standard deviation, and the sample size, that is the number of publications which list the number of nodes, is 38. The dashed line is plotted at $y = 4$ and crosses almost all error bars, which indicates that the number of nodes through the years stayed stable.

There is one large outlier in 2005, one of the publications [2] mentioned that they have tested their system with 30 nodes (and a master).

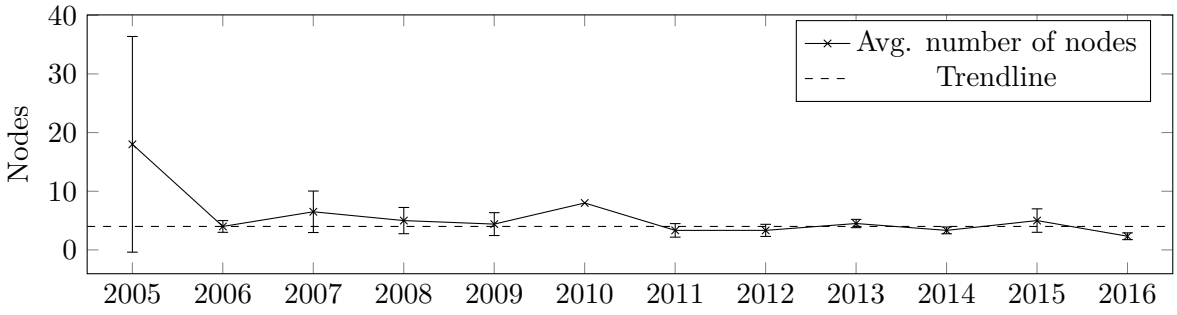


Figure 5: Average number of nodes per year (sample size: 38)

4 Topology

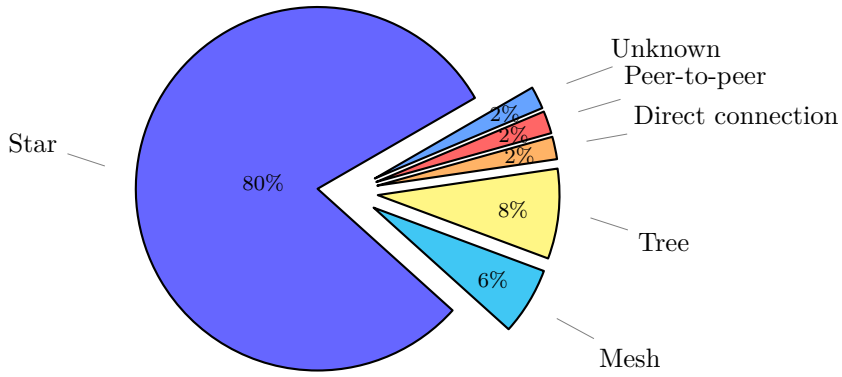


Figure 6: Topologies used by WBAN systems (sample size: 53)

Unsurprisingly there is a clear winner regarding the topologies of the reviewed WBAN systems. About 80% of the systems use a plain star topology, and 6% use a tree approach, which we can see as a multi-hop star topology. The tree topology makes for more flexible communication as relay nodes can be used for increasing the reliability of the communication. E.g. [21] uses a tree topology to relay data from implants to the central node, to ensure a good communication link, because the signal is less able to propagate through the body mass. The relays are placed on the skin at the location of the implants.

The large advantage of a star topology is its simplicity and the fact that a routing protocol is not needed: all nodes have a direct connection to the central node, after all. Also, the most

well-known protocols only support a star topology such as Bluetooth and WiFi. Zigbee has also support for multi-hop communication to be able to form a tree topology, besides the support for star networks.

A disadvantage of a star network is the single point of failure property: if the central node stops working the whole system is unable to operate. The same counts for a tree network, because there is also one central node which collects the data and provides the communication link to beyond-BAN networks.

Other topologies used by the reviewed papers are:

- **Mesh** (3 systems): In a mesh network nodes have the ability to connect to every node directly. More complex routing protocols are needed to route data to a sink node.
- **Peer-to-peer** (1 system): In a peer-to-peer network every node is an endpoint and has it's own responsibilities. If data has to be sent to a certain endpoint, the node has to make a connection itself and send the data. There is no central or sink node which takes care of the data transport and beyond-BAN communication. Although one of the end nodes could have this responsibility.
- **Direct connection** (1 system): A direct connection actually means the lack of a topology because just two nodes are being connected to each other.

That so many systems choose for a star topology is not a surprise. A star network is the easiest to set up because its an established network topology, all used wireless protocols support it and no routing protocol is needed. Also a star topology has the smallest delays, because packages do not have to be relayed. For most researchers it is the default choice: other topologies do require further research regarding implementation and design choices while this is not necessary for a star topology.

The second choice is a tree topology which is used by four systems. The main consideration for choosing a tree network is reliability. The distances between nodes become smaller if one adds relay nodes, as is the case in a tree network.

5 Wireless Protocols

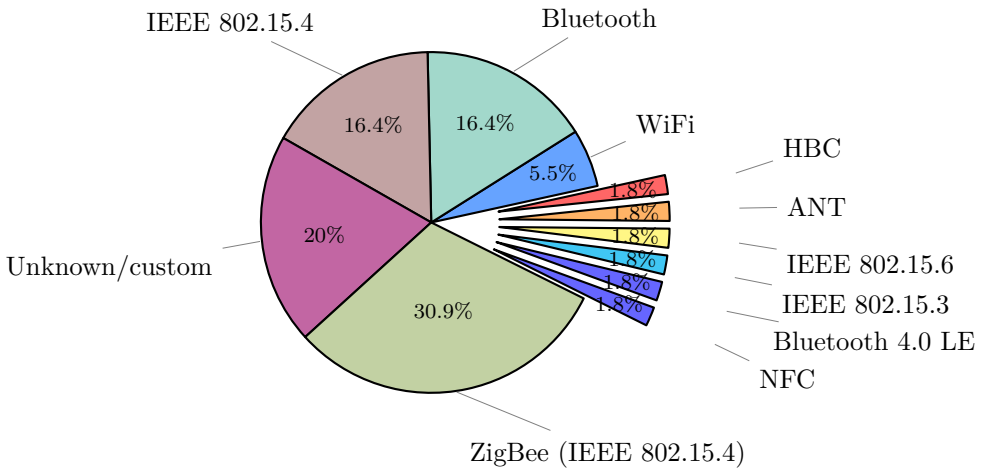


Figure 7: Used protocols by WBAN systems (sample size: 55)

Several standards exist that can be used as the wireless protocol in WBANs. The most well-known ones are Bluetooth, Zigbee and WiFi. In addition to those IEEE has published a new standard specialised for WBAN communication, which is IEEE 802.15.6 [53]. This protocol, however, is still not used by a lot of the researched systems, as there are not a lot of off-the-shelf components and libraries that support it. Only one system has used a protocol based on IEEE 802.15.6, see Figure 7. The problem is also that only the Physical and Medium Access Control (MAC) layer are defined by IEEE, while the Networking and Application layers have to be implemented by the developer himself. This takes more effort and research to come to a good working solution.

Some systems use more than one protocol, e.g. [54] uses a form of Human Body Communication, ZigBee and NFC. These protocols are listed separately in Figure 7 and therefore the sample size exceeds the number of surveyed systems.

Further we see that about 20% of the systems do not list or specify the protocol they use. It is most likely that they didn't use a standard, and created their own protocol. The advantage of creating a custom protocol is that it can be relative lightweight and targeted at the application of the developed WBAN. The disadvantage is that there is no interoperability possible between different systems, or the nodes from different systems.

The most popular choices for WBANs are IEEE 802.15.4/ZigBee and Bluetooth. The advantage of these protocols is that they are well known and therefore already have a lot of hardware, software and community support available. If the goal of a project is to prove the feasibility of a certain application with the use of a WBAN, it is a good choice to choose an established protocol. Also WiFi is used in some systems, but is not a popular choice due to the higher power consumption.

A remarkable fact is that most systems use Bluetooth version 3 or lower instead of the more energy efficient Bluetooth 4 Low Energy. Only one system has specified that they use the Low Energy variant of Bluetooth. We have to add that not all publications have defined the Bluetooth version they used, and that some of the older publication have not been able to use Bluetooth 4 as it was released in 2010. ZigBee is the most widely used protocol, and has a lower energy consumption and supports multi-hop networks. ZigBee is based on the IEEE 802.15.4 standard created by IEEE, and defines an network and application layer on top of this protocol. Some WBAN systems do use the IEEE 802.15.4 standard, but not the higher layers as defined by ZigBee, probably to keep the communication overhead low while being able to use ZigBee hardware.

6 Frequency bands and Physical layer

The used frequencies are highly related to the used wireless protocols. Most of the protocols only support one frequency band, e.g. WiFi and Bluetooth only support the 2.4GHz band. ZigBee is also mainly used in the 2.4GHz band, while additionally also having support for the 868MHz and 915MHz bands. 2.4GHz is therefore the most used band: 80 percent of the surveyed systems use the 2.4GHz band, see also Figure 8.

Ultra Wide Band (UWB) follows with a usage in 12% of the systems. The frequency range used for UWB can be very broad and differs per system. The allowed range is 3.1GHz up to 10.6GHz, although with a maximum power spectral density (PSD) of -41 dBm/MHz. A big advantage of using the ultra wide band is that it allows for high data rates of up to 20Mbps, while maintaining a high energy efficiency. This data rate can be necessary for sensors which monitor multiple channels or high frequency signals e.g. signals such as ECG, ECC or EMG. Also video streams can be transported using this link, which can be useful for capsule endoscopy. UWB is shown as feasible communication link for WBANs in [55] and used as main wireless

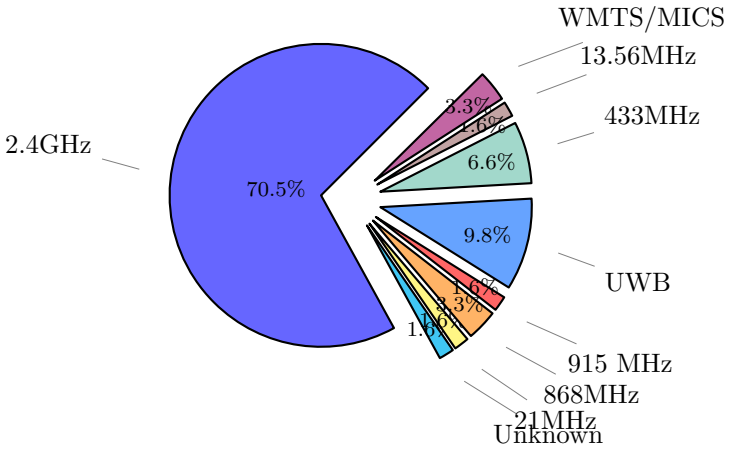


Figure 8: Used frequencies by the WBAN systems

communication link in e.g. [4], [10] and [21].

Human Body Communication (HBC) is only used by one of the systems as addition to 2.4GHz. But HBC is defined as one of the possible communication links in IEEE 802.15.6 and has some unique features which are interesting for creating a secure and reliable WBAN as stated in [56]:

- **Secure barrier**: A HBC network is enclosed by the body, and therefore it is not possible to eavesdrop communication between nodes remotely. The required communication frequency for HBC is much lower than for Narrow Band (NB) or UWB communication, at which the wavelength is much longer than the body's channel length, so the body does not work as antenna.
- **Less interference**: Because the signal does not propagate outside the body, and the used frequencies do not match those of the crowded RF bands, interference is very low.
- **Energy consumption**: As shown by the authors of [57], HBC uses an order of magnitude less energy per bit than UWB, which is already very energy efficient. ZigBee uses around 106nJ/b, Bluetooth 2.1 uses 11.9nJ/b, UWB 2.5nJ/b while HBC gets down to 0.24nJ/b.
- **Data rate**: The offered data rate of up to 10 Mb/s is quite high compared to ZigBee and Bluetooth, but lower than UWB which offers data rates up to 20MB/s.

A disadvantage of HBC compared to RF communication is that it is required to have close contact with the body. A movement tracking bracelet e.g. does not always have proper skin contact.

In Figure 9 we compare the different frequency bands and also include Human Body Communication (HBC) regarding the energy efficiency, interference levels and possible data rates. HBC seems like a good solution because of low interference, high energy efficiency and a high possible data rate.

7 Channel Access

Also the channel access mechanisms are related to the used protocols. Some protocols support more than one channel access protocol, but others have defined one particular protocol which

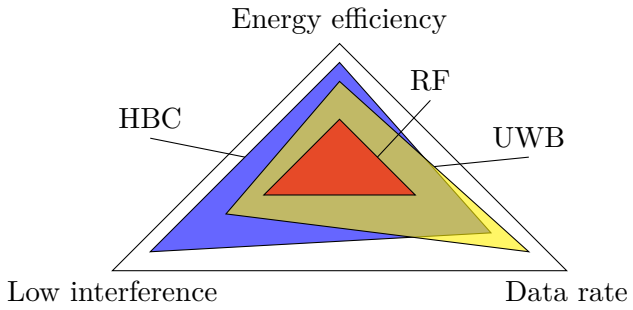


Figure 9: Comparison of interference, efficiency and data rate regarding the physical layer

has to be used. The Channel Access Method is related to Medium Access Control protocols, but these are in general also concerned with addressing and assigning channels to users.

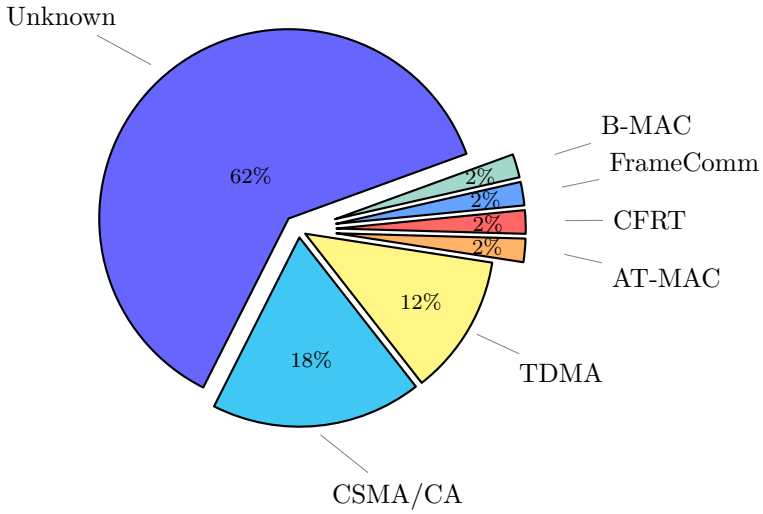


Figure 10: Channel access mechanisms of the surveyed systems

The distribution of the used channel access protocols is quite different from the previous discussed items: most systems do not define the channel access protocol they use. It is also not possible to derive the used channel access method from the use wireless protocols, as this is not always fixed per protocol, see Table 2. The most popular choices are Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and Time Division Multiple Access (TDMA). Because IEEE 802.15.4/ZigBee and Bluetooth also mainly use channel access mechanisms based on CSMA/CA and TDMA, we can assume that the largest part of the systems from which we do not know the channel access mechanism will use either CSMA/CA or TDMA.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) uses a mechanism to sense whether another node is already using the wireless channel, and tries to avoid any collisions, by waiting for a random short time in case a carrier is sensed, before sending it's data. When the cumulative data rate on a given network increases, the chance for a collision also increases because the chance that two nodes start sending at the same time increases. An advantage of CSMA/CA is that, at least for networks with not too much traffic, the delays are very small: a node does never have to wait for an assigned time slot which is better for applications with tight timing constraints.

Time Division Multiple Access (TDMA) uses assigned time slots for each node at which they can use the channel. An advantage is that the channel will be free during the assigned time slot,

Wireless protocol	Possible Channel Access protocols
IEEE 802.15.6	Slotted ALOHA, CSMA/CA
IEEE 802.15.4/ZigBee	Unslotted CSMA/CA, Beaconed TDMA
Bluetooth	CSMA/CA, TDMA
Wifi	CSMA/CA

Table 2: Wireless protocols with their possible channel access methods

so there are no collisions, and therefore the amount of wasted energy is minimised. Another advantage is that a node can wake-up at this certain time slot, and sleep for the rest of the period, so for high data rate networks it is more energy efficient. Some systems send out beacons to synchronise the time slot schedule and to synchronise timers. For data fusion purposes the relative measurement time of signals can be very important, otherwise synchronisation of the signals itself becomes imprecise.

The rest of the channel access protocols we came across are actually a variation on TDMA or CSMA/CA. The protocols are:

- **CFRT**: The authors of [16] have created a MAC protocol which they name the Collision Free Real Time (CFRT) protocol. But it is actually almost the same as TDMA with superframes.
- **AT-MAC**: Adaptive Data Transmission MAC (AT-MAC) [58] is another protocol used by one of the WBAN systems, and is actually TDMA with flexible assignable time slots for each node by a master.
- **FrameComm**: FrameComm [59] is a duty-cycled CSMA protocol which works with packet bursts, also called framelets, to send data or synchronisation packages. The framelets which span the entire duty-cycle are needed because the receiver is only listening during a certain period of the time, and data could be lost otherwise.
- **B-MAC** [60]: Is a CSMA-like protocol with a custom Clear Channel Assesment (CCA) method.

In general a WBAN requires a channel access protocol that can handle both strict periodic data, high priority data and a high data rate. A lot of sensors that measure a certain physical signal will report this periodically, but will not have an excessive high data rate. But there are also nodes that do need a high data rate, e.g. to be able to stream video data, although a video stream is generally also periodic. These requirements advocate more for a TDMA like protocol. There is only one caveat regarding high priority events. The transmission delay should be as low as possible, which is not the case if a node has to wait for it's assigned slot if TDMA is used. With CSMA/CA a node can send directly after assessing whether the channel is clear, and makes for lower delays. Therefore a hybrid approach seems the best way to go. In Figure 11 one can see the TDMA and CSMA/CA protocols compared regarding average packet delays, energy efficiency and data rate.

8 Routing protocols

The list of used routing protocols in the researched systems is far from abundant. Due to the fact that most systems use a star or tree topology, for which a routing protocol is not needed, only three systems use a routing protocol:

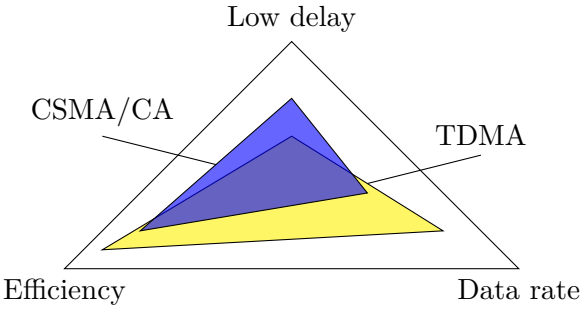


Figure 11: Comparison of package delay, energy efficiency and data rate regarding channel access methods

- **Publish/subscribe routing**: The CodeBlue WBAN system as described in [2] uses a publish/subscribe routing protocol for multi-hop networks. Nodes publish data to a certain channel, and end-user devices can subscribe to this channel to receive the data. Multi-hop routing data is maintained in a table at every node, and updated if a lower cost path is encountered. A discovery protocol is used to discover new nodes, where nodes need to broadcast their identity. Data from a certain channel is transmitted to all the subscribed nodes.
- **Flows** [8]: This protocol uses almost the same approach as the one used in the CodeBlue project, but allows for multiple coexisting spanning trees that span a certain set of nodes, where the CodeBlue project uses only one spanning tree to span all nodes in the network.
- **Custom implementation**: The WBAN project from [40] describes a custom routing implementation for their mesh network. A node can auto-configure itself as router and will then be updated by the coordinator with the proper routing information. Also end-user devices will be updated with the needed routing information by it's parent, which can be a router or the coordinator. If the parent is a router, it will then notify the coordinator about the new end-user device. Data packets are routed based on the routing tables in each device.

Apart from these three protocols other routing protocols have been developed especially for WBANs. Some have as goal to be as energy efficient as possible, others keep track of the radiated energy which should not exceed a certain level as regulations describe. Routing protocols seem to be only applicable to more complex networks: in a star network every node has a direct connection with the central node and nodes are only concerned with channel access and prevention of collisions.

9 Energy supply

All nodes of the surveyed systems use batteries as energy supply, but most publications do not mention the battery life. There can be at least two possible reasons: the researchers did not optimise their system on energy efficiency and therefore do not want to publish anything about it, or the battery life was still so bad that they did not want to mention it.

Only four papers mention the battery life of their nodes. These vary between 10 and 63 hours, which is by far not enough for a proper WBAN. Replacing or recharging batteries of all nodes each one to two and a half day is not very practical. More research effort is needed to create real low-power nodes.

However only one paper mentions using energy harvesting [6], a lot of research has been done on the field of energy harvesting from the human body. The only surveyed system which mentions energy scavenging is [6] and reports a generated amount of power of 0.1mW. [61] gives an in-depth overview of all the available methods to harvest energy in general, while [62] only lists the most common methods to harvest energy from a human body, of which we give a short overview here:

- **Thermoelectric generators**: A thermoelectric generator uses body heat to generate electrical energy. The problem with these types of generators is that a temperature difference is needed to generate the power, so insulating clothing or a high ambient temperature influences the generated amount of power negatively. [63] reports an average power production of 10-30$\mu W/cm^2$ for a typical person indoors.
- **Piezoelectric generators**: It's possible to generate electrical energy by using a piezoelectric material. When a piezoelectric material is deformed due to mechanical force, a voltage is generated. The amount of energy depends on the piezoelectric properties of the used material and the amount of deformation applied to the material. This method has mainly been used in shoes, because it is easy to deform a material when walking. The reported amount of generated power varies from 1.3mW up to 8.3mW.
- **Electromagnetic generators**: Another way to use kinetic energy from the body is by using electromagnetic generators. A number of options are listed in [64], which use either a spring with a magnet to transform vibrations into electrical energy or a rotor to transform rotational energy into electrical energy. Larger systems are also built, e.g. using a lever attached to the upper part of the leg, so rotational movements of the hip joint can be used for energy generation.

An example of a battery-less ExG node is described in [65] which makes use of a Thermal Energy Generator (TEG). Another example of a hybrid way of scavenging energy is given in [66] where piezoelectric and electromagnetic generators are combined.

There are also other ways of charging a node without replacing batteries, which include wireless charging techniques. We will give a short overview of some methods:

- **Resonating magnetic field**: This is the technique used by modern wireless phone chargers. The technique is based on an oscillating magnetic field, which is generated by the sending coil, and induces a current in the receiving coil if the magnetic field passes through this coil. By using magnetic resonance is is possible to couple the two coils and get a high efficiency. The authors of [67] take this even a step further and propose a system where beam forming is used to charge multiple devices at a time, at a distance of up to 50cm.
- **Ultrasound**: Energy can also be transferred using ultrasound, and also here resonance is used to increase the efficiency [68]. The energy is propagated through tissue as mechanical vibration energy and acoustic energy, which is transformed back to electrical energy at the receiver side. The maximum efficiency is just over 22% through 10mm skin tissue.

10 Security

Security is an important aspect of WBANs due to the fact that the involved information is strict personal and important. Especially if the information is used by other nodes to perform an action based on the measured data, the integrity of the data is very important. The following aspects are required to have a secure WBAN:

- **Authentication**: Authentication is one of the basic security measures applied to all sorts of networks and applications, and implies verification and identification of the sender of the data. Data can only be trusted if the sender is to be known as being not a false adversary.
- **Authorisation**: Authorisation means that the controller can allow or disallow a certain action of a authenticated node in the network. Authorisation mechanisms are not always available in WBAN networks because there is not always control needed of the actions performed by a node in the network.
- **Integrity**: Data integrity implies the verification of the data whether it is not altered during transmission. To prove this, together with authentication, a Message Authentication Code (MAC) or Message Integrety Code (MIC) is often used.
- **Confidentiality**: Data confidentiality is needed to prevent disclosure of data. Data can be obtained by using eavesdropping or overhearing, and even with proper authentication and data integrity mechanisms this data can be valuable. Data confidentiality normally means applying encryption to the data.
- **Secure key management**: Keys are used for encryption, authentication and integrity, so when they can be obtained by an untrusted source, they can pretend to be a trusted party and decrypt all data. Therefore keys should be managed and stored in a secure way on all nodes.
- **Availability**: The availability of the nodes in a network should be guaranteed at all times. Some information can be very important, and lives can depend on it. Attackers can perform all kinds of Denial of Service (DoS) attacks to get a system down, e.g. by jamming on the used communication frequency band.
- **Data Freshness**: Data freshness techniques imply measures against replaying of packets. If a non-trusted node can capture a packet and replay this, without altering it data, it will still be seen as a authorised and integer package, while the data is outdated and can harm or disrupt proper operation of the system.
- **Random Number Generator**: Encryption keys are created based on a random number which is generally generated by a Random Number Generator (RNG). The entropy of such a RNG is important because if it is very low, the random number and therefore the generated key can be guessed by an attacker.

Only three of the surveyed systems indicate the cipher suite they use for encryption, in all other papers, encryption is not mentioned. Although all systems that use ZigBee, Bluetooth, Wifi, IEEE 802.15.3 or IEEE 802.15.6 already have to use some sort of security measures, so how secure the communication really is depends on implementation choices and how a system is set up. In the following subsections we will give a short overview of the security aspects of Bluetooth, ZigBee and IEEE 802.15.6 and some of the known vulnerabilities.

10.1 Bluetooth

An analysis of Bluetooth (BT) security and it vulnerabilities can be found in [69], of which we will give a short overview. Each BT device possesses a unique identification number: a 48-bit address, furhter a private 128-bit authentication key and a 8 to 128-bit private key depending on the required level of security. Also every BT device has a 128-bit random number which is used as

seed for the random number generator. Several other keys are used during communication, and are generated per device-pair or on a session basis. There are several encryption modes available depending on whether a device uses a semi-permanent link key or a master key, varying from no encryption at all, all data except broadcast messages are encrypted or all traffic is encrypted. Bluetooth uses a challenge-response strategy as authentication scheme with as result that both participants have the same symmetric key.

There are several vulnerabilities known to attack a Bluetooth system. With a man-in-the-middle attack the keys exchanged during authentication can be obtained. Which then can be used to impersonate a node, or eavesdrop on communications. Another known vulnerability is spoofing the BT address. As nodes already put some trust in the identity of the device based on the address, one can impersonate a device by changing it's address to the address of another device. Also the PIN length makes for another vulnerability, as a lot of devices use a very short PIN of only 4 digits. An attacker can use a brute-force attack using all possible PIN's to get to the proper key.

Other vulnerabilities relate to improper validation of BT implementations, exposure and potential for improper randomness. The level of security highly relates to the implementation and communication choices. Take e.g. the private key length which can vary from 8 to 128 bit, the pin length and encryption mode which can be choosen by the application.

10.2 ZigBee

ZigBee, based on IEEE 802.15.4 [70], which defines only the physical and MAC layers, uses several security measures to ensure secure communication. It utilises the CCM* encryption algorithm which is based on 128-bit AES encryption in Cipher Block Chaining (CBC) mode and offers measures for authentication, data integrity and sequential data freshness [71]. Data integrity is provided using CBC-MAC, an encrypt-then-MAC protocol which provides an integrity check on the ciphertext. ZigBee has a network manager, a trust manager and a configuration manager which are responsible for the key distribution, device authentication and end-to-end security respectively. A network can operate in two modes: residential and commercial mode. Residential mode offers a lower security level and means that all nodes use the same key for encryption: the network key K_N. This is not very secure because it is not possible to prevent insider attacks. It is however better resource-optimised because less memory and bandwidth are needed for the key exchange. Commercial mode offers a higher security level and applies a separate encryption key for every device-pair in the network.

Also ZigBee has a vulnerability regarding initial key negotiation [72]. AES uses symmetrical encryption keys, so with a man-in-the-middle attack it is possible to obtain the key and impersonate a node, especially when a new node joins the network. Another vulnerability includes that it is possible to use ZigBee without integrity check and with only encryption, or with only integrity check and no encryption. This is dependent on the developers implementation. Also ZigBee network addresses are dynamic and handed out by the network controller. If a node or network controller leaves and joins the network again, nodes can have double addresses or get a new address. Using dynamic addresses and on-demand routing makes it easier to impersonate a node.

10.3 IEEE 802.15.6

IEEE 802.15.6 [53] uses a security hierarchy of thee layers. Using authentication credentials a master key (MK) is generated for communication between two parties. Once per session a Pairwise Temporal Key (PTK) is created which is used for communication during a session. A session's duration is determined by the security policy on which the communicating parties have

to agree. Once per message (or frame) authentity verification, encryption and replay defence measures are applied.

IEEE 802.15.6 offers three security levels:

- **Level 0**: unsecured communication. All messages are transmitted without authentication, integrity validation, confidentiality and replay defence.
- **Level 1**: authentication but no encryption. Messages are transmitted with authentication and integrity validation and replay defence, but without confidentiality protection.
- **Level 2**: authentication and encryption. All messages are transmitted with authentication, integrity validation, confidentiality and replay defence.

The node and a hub jointly have to select a security level during association. For unicast secured communication, the node and hub need to activate a pre-shared MK or establish a new MK during the association phase. This is done using a protocol based on Diffie-Helmann Key exchange and elliptic curve public key cryptography. In total seven algorithms are used in the IEEE 802.15.6 standard for ensuring secure communication.

IEEE 802.15.6 is vulnerable for Key Compromise Impersonation (KCI) attacks. Using the address of another node it's possible to agree with the network hub on a master key while impersonating this node. This can be prevented by using public key certificates, but then it is still possible to impersonate the hub [73]. Most algorithms also lack forward secrecy, which means that an attacker can obtain enough data from the packages to calculate the master key used for communication by two other nodes, and decrypt all previously recorded packages. Forward secrecy means that if the communication key is compromised, it cannot be used to reveal any data sent before.

11 Discussion

Looking at the last ten years of research we see that Wireless Body Area Networks (WBANs) become an established phenomenon. The initial hype passed, but more research is published at a stable rate. The results of this paper are based on over 50 publications of complete WBAN systems.

The majority of these systems use a simple star topology, and only a few use a mesh or tree topology. Connected to this we see that routing protocols are not used a lot, because it's not needed for a star network. The most popular protocols are IEEE 802.15.4/ZigBee and Bluetooth, which are mainly used in the 2.4GHz band. The second choice for a physical layer is Ultra Wide Band communication. Surprisingly, only one system uses Human Body Communication (HBC) while it offers promising properties for WBANs. Channel access protocols are limited to variations of just two protocols: Carries Sense Multiple Access with Collision Avoidance (CSMA/CA) and Time Division Multiple Access (TDMA). CSMA/CA is better for low-rate applications with low package delay requirements, while TDMA seems to be better for high data-rate applications.

Security is an important aspect of WBANs due to the sensitive and personal information it processes. The main established protocols do not offer total security, as several threats are known. IEEE 802.15.6 offers better security than Bluetooth and ZigBee because initial key negotiation is done using a asymmetric key exchange, but IEEE 802.15.6 is still vulnerable for impersonation attacks and does not offer forward secrecy.

As a result of the reported state of research we see several fields on which more research is needed. In general more research is needed to make a WBAN more mature an resilience,

for which we need better energy efficiency, better interoperability and better security. Also research is needed on Human Body Communication because it is a promising technique which is not applied a lot to real WBAN systems.

References

[1] T. Norgall, R. Schmidt, and T. Von Der Grun, "Body area network: a key infrastructure element for patient-centered telemedicine," *Biomedizinische Technik/Biomedical Engineering*, vol. 47, no. s1a, pp. 365–368, 2002.

[2] V. Shnayder, B.-r. Chen, K. Lorincz, T. R. F. F. Jones, and M. Welsh, "Sensor networks for Medical Care," *Proceedings of Third International Conference on Embedded Networked Sensor Systems*, no. June, p. 314, 2005.

[3] E. Jovanov, A. Milenkovic, C. Otto, and P. C. de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 2, no. 1, pp. 1–10, 2005.

[4] T. Falck, J. Espina, J. P. Ebert, and D. Dietterle, "BASUMA - The sixth sense for chronically III patients," *Proceedings - BSN 2006: International Workshop on Wearable and Implantable Body Sensor Networks*, vol. 2006, pp. 57–60, 2006.

[5] F. E. H. Tay, M. N. Nyan, D. G. Guo, C. P. Ng, L. Xu, and C. T. Tan, "MEMSWear - Biomonitoring system in a body area network (BAN)," *Proceedings of the International Semiconductor Conference, CAS*, vol. 1, pp. 207–214, 2007.

[6] B. Gyselinckx, R. Vullers, C. Hoof, J. Ryckaert, R. Yazicioglu, P. Fiorini, and V. Leonov, "Human++: Emerging Technology for Body Area Networks," in *2006 IFIP International Conference on Very Large Scale Integration*, vol. 249, (Boston, MA), pp. 175–180, IEEE, oct 2006.

[7] C. Otto and a. Milenkovic, "System architecture of a wireless body area sensor network for ubiquitous health monitoring," *Journal of Mobile ...*, vol. 1, no. 4, pp. 307–326, 2006.

[8] T. Gao, T. Massey, L. Selavo, D. Crawford, B. R. Chen, K. Lorincz, V. Shnayder, L. Hauenstein, F. Dabiri, J. Jeng, A. Chanmugam, D. White, M. Sarrafzadeh, and M. Welsh, "The advanced health and disaster aid network: A light-weight wireless medical system for triage," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 1, no. 3, pp. 203–216, 2007.

[9] L. Wolf and S. Saadaoui, "Architecture Concept of a Wireless Body Area Sensor Network for Health Monitoring of Elderly People," *2007 4th IEEE Consumer Communications and Networking Conference*, pp. 722–726, 2007.

[10] D. Domenicali and M.-G. D. Benedetto, "Performance Analysis for a Body Area Network composed of IEEE 802.15.4a devices," *4th Workshop on Positioning, Navigation and Communication*, vol. 2007, pp. 273–276, 2007.

[11] N. de Vicq, F. Robert, J. Penders, B. Gyselinckx, and T. Torfs, "Wireless Body Area Network for Sleep Staging," *Biomedical Circuits and Systems Conference*, pp. 163–166, 2007.

[12] E. Monton, J. Hernandez, J. Blasco, T. Herve, J. Micallef, I. Grech, A. Brincat, and V. Traver, "Body area network for wireless patient monitoring," *IET Communications*, vol. 2, no. 2, p. 215, 2008.

[13] M. Soini, J. Nummela, and P. Oksa, "Wireless body area network for hip rehabilitation system," *Ubiquitous Computing . . .*, vol. 3, no. 5, pp. 42–48, 2008.

[14] Li Xuemei, Jiang Liangzhong, and Li Jincheng, "Home healthcare platform based on wireless sensor networks," in *2008 International Conference on Technology and Applications in Biomedicine*, pp. 263–266, IEEE, may 2008.

[15] M. Hamel, R. Fontaine, and P. Boissy, "In-home telerehabilitation for geriatric patients," *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, pp. 29–37, jul 2008.

[16] E. Farella, A. Pieracci, L. Benini, L. Rocchi, and A. Acquaviva, "Interfacing human and computer with wireless body area sensor networks: The WiMoCA solution," *Multimedia Tools and Applications*, vol. 38, no. 3, pp. 337–363, 2008.

[17] S. Jiang, Y. Cao, S. Iyengar, P. Kuryloski, R. Jafari, Y. Xue, R. Bajcsy, and S. Wicker, "CareNet: An Integrated Wireless Sensor Networking Environment for Remote Healthcare," *ACM Transactions on Embedded Computing Systems*, p. 3, 2009.

[18] M. Sukor, S. Ariffin, N. Fisal, S. S. S. Yusof, and A. Abdallah, "Performance Study of Wireless Body Area Network in Medical Environment," *Modeling {&} Simulation, 2008. AICMS 08. Second Asia International Conference on*, pp. 202–206, 2008.

[19] R. A. Rashid, S. H. S. Arifin, M. R. A. Rahim, M. A. Sarijari, and N. H. Mahalin, "Home healthcare via wireless biomedical sensor network," *2008 IEEE International RF and Microwave Conference, RFM 2008*, pp. 511–514, 2008.

[20] J. Y. Khan, M. R. Yuce, and F. Karami, "Performance evaluation of a Wireless Body Area sensor network for remote patient monitoring.," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2008, pp. 1266–1269, 2008.

[21] R. Chávez-Santiago, "Architecture of an ultra wideband wireless body area network for medical applications," *Applied Sciences in . . .*, vol. 1, pp. 1 – 6, 2009.

[22] B. Wang, L. Wang, S. J. Lin, D. Wu, B. Y. Huang, Y. T. Zhang, Q. Yin, and W. Chen, "A Body Sensor Networks Development Platform for Pervasive Healthcare," *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, pp. 1–4, 2009.

[23] A. Saeed, M. Faezipour, M. Nourani, S. Banerjee, G. Lee, G. Gupta, and L. Tamil, "A Scalable Wireless Body Area Network for Bio-Telemetry," *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 77–86, 2009.

[24] L. Brown, B. Grundlehner, J. van de Molengraft, J. Penders, and B. Gyselinckx, "Body area network for monitoring autonomic nervous system responses," *Proceedings of the 3d International ICST Conference on Pervasive Computing Technologies for Healthcare*, pp. 10–12, 2009.

[25] A. T. Barth, M. A. Hanson, H. C. Powell, and J. Lach, "TEMPO 3.1: A body area sensor network platform for continuous movement assessment," *Proceedings - 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009*, pp. 71–76, 2009.

[26] T. O'Donovan, J. O'Donoghue, C. Sreenan, D. Sammon, P. O'Reilly, and K. O'Connor, "A context aware wireless body area network (BAN)," *2009 3rd International Conference on Pervasive Computing Technologies for Healthcare*, pp. 1–8, 2009.

[27] K. Wac, R. Bults, B. van Beijnum, I. Widya, V. Jones, D. Konstantas, M. Vollenbroek-Hutten, and H. Hermens, "Mobile patient monitoring: The MobiHealth system," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1238–1241, IEEE, sep 2009.

[28] M. R. Yuce, "Implementation of wireless body area networks for healthcare systems," *Sensors and Actuators, A: Physical*, vol. 162, no. 1, pp. 116–129, 2010.

[29] E. Kantoch, M. Smolen, P. Augustyniak, and P. Kowalski, "Wireless body area network system based on ECG and accelerometer pattern," *Computing in Cardiology*, pp. 245–248, 2011.

[30] S. Sharma, A. L. Vyas, B. Thakker, D. Mulvaney, and S. Datta, "Wireless Body Area Network for health monitoring," *2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*, pp. 2183–2186, 2011.

[31] R. Marin-Perianu, M. Marin-Perianu, P. Havinga, S. Taylor, R. Begg, M. Palaniswami, and D. Rouffet, "A performance analysis of a wireless body-area network monitoring system for professional cycling," *Personal and Ubiquitous Computing*, vol. 17, no. 1, pp. 197–209, 2013.

[32] A. Nassir and O. Barnea, "Wireless body-area network for detection of sleep disorders," *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel, IEEEI 2012*, pp. 1–5, 2012.

[33] M. Wagner and B. Kuch, "Android based Body Area Network for the evaluation of medical parameters," *10th International Workshop on Intelligent Solutions in Embedded Systems*, pp. 33–38, 2012.

[34] C. Wang, Q. Wang, and S. Shi, "A distributed wireless body area network for medical supervision," *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 2612–2616, 2012.

[35] U. Mitra, B. A. Emken, Sangwon Lee, Ming Li, V. Rozgic, G. Thatte, H. Vathsangam, D. Zois, M. Annavaram, S. Narayanan, M. Levorato, D. Spruijt-Metz, and G. Sukhatme, "KNOWME: a case study in wireless body area sensor network design," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 116–125, 2012.

[36] Y. Hamada, K. Takizawa, and T. Ikegami, "Highly reliable wireless body area network using error correcting codes," *2012 IEEE Radio and Wireless Symposium*, pp. 231–234, 2012.

[37] K. Li and S. Warren, "High resolution wireless body area network with statistically synchronized sensor data for tracking pulse wave velocity," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 2080–2083, 2012.

[38] S.-L. Tan, J. Garcia-Guzman, and F.-H. Villa-Lopez, "A wireless body area network for pervasive health monitoring within smart environments," *2012 IEEE Second International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, pp. 47–51, 2012.

[39] P. Dinkar, A. Gulavani, S. Ketkale, P. Kadam, and S. Dabhade, "Remote Health Monitoring using Wireless Body Area Network," no. 4, pp. 399–402, 2013.

[40] Y. Varatharajah, N. Karunathilaka, M. Rismi, S. Kotinkaduwa, and D. Dias, "Body area sensor network for evaluating fitness exercise," *Proceedings of 2013 6th Joint IFIP Wireless and Mobile Networking Conference, WMNC 2013*, 2013.

[41] Y. Shi and Y. Zhang, "Smartphone based body area network system," *Proceedings - 2014 International Conference on Medical Biometrics, ICMB 2014*, pp. 204–209, 2014.

[42] M. Chen, Z. Li, and G. Zhang, "A cooperative software-hardware approach for wireless body area network implementation," in *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*, pp. 214–218, IEEE, jun 2014.

[43] Z. He and X. Bai, "A wearable wireless body area network for human activity recognition," *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 115–119, 2014.

[44] U. Ghoshdastider, R. Viga, and M. Kraft, "Non-invasive synchronized spatially high-resolution wireless body area network," in *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pp. 1–6, IEEE, apr 2014.

[45] K. Thotahewa, J.-M. Redoute, and M. R. Yuce, "A Low-Power Wearable Dual-Band Wireless Body Area Network System: Development and Experimental Evaluation," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 62, no. 11, pp. 2802–2811, 2014.

[46] B. R. Nandkishor and A. B. A. N. Architecture, "Android Smartphone Based Body Area Network for Monitoring and Evaluation of Medical Parameters," pp. 284–288, 2014.

[47] Z. Dong, H. Gu, Y. Wan, W. Zhuang, R. Rojas-Cessa, and E. Rabin, "Wireless body area sensor network for posture and gait monitoring of individuals with Parkinson's disease," *ICNSC 2015 - 2015 IEEE 12th International Conference on Networking, Sensing and Control*, pp. 81–86, 2015.

[48] T. Przybylski, P. Froehle, C. Mcdonald, M. Mirzaee, S. Noghanian, R. Fazel-rezai, and A. E. Hardware, "Wearable Wireless Body Area Network for Aeronautical Applications," pp. 563–568, 2015.

[49] J. A. Hidalgo, A. Cajiao, C. M. Hernández, D. M. López, and V. M. Quintero, "VISIGNET: A wireless body area network with cloud data storage for the telemonitoring of vital signs," *Health and Technology*, vol. 5, no. 2, pp. 115–126, 2015.

[50] G. Wolgast, C. Ehrenborg, A. Israelsson, J. Helander, E. Johansson, and H. Manefjord, "Wireless Body Area Network for Heart Attack Detection [Education Corner]," *IEEE Antennas and Propagation Magazine*, vol. 58, no. 5, 2016.

[51] R. Li, Z. Cai, W. Lee, and D. T. H. Lai, "A Wearable Biofeedback Control System Based Body Area Network for Freestyle Swimming," pp. 1866–1869, 2016.

[52] Y. Wang, Y. Zheng, O. Bai, Q. Wang, D. Liu, X. Liu, and J. Sun, "A multifunctional wireless body area sensors network with real time embedded data analysis," in *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 508–511, IEEE, oct 2016.

[53] IEEE Standards Association, *IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks*. No. February, 2012.

[54] N. Fahier and W.-C. Fang, "An advanced plug-and-play network architecture for wireless body area network using HBC, Zigbee and NFC," in *2014 IEEE International Conference on Consumer Electronics - Taiwan*, pp. 165–166, IEEE, may 2014.

[55] M. R. Yuce, H. C. Keong, and M. S. Chae, "Wideband communication for implantable and wearable systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 57, no. 10, pp. 2597–2604, 2009.

[56] M. Seyedi, S. Member, B. Kibret, S. Member, D. T. H. Lai, and M. Faulkner, "A Survey on Intrabody Communications for Body Area Network Applications," vol. 60, no. 8, pp. 2067–2079, 2013.

[57] J. Bae, S. Member, K. Song, and S. Member, "A Low-Energy Crystal-Less Double-FSK Sensor Node Transceiver for Wireless Body-Area Network," vol. 47, no. 11, pp. 2678–2692, 2012.

[58] Z. Li, G. Zhang, and W. J. Li, "An adaptive data transmission scheme for Wireless Body Area Networks," *2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 399–403, 2012.

[59] J. Benson, T. O'Donovan, U. Roedig, and C. J. Sreenan, "Opportunistic aggregation over duty cycled communications in wireless sensor networks," *Proceedings - 2008 International Conference on Information Processing in Sensor Networks, IPSN 2008*, pp. 307–318, 2008.

[60] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 95–107, 2004.

[61] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green, "Energy harvesting from human and machine motion for wireless electronic devices," *Proceedings of the IEEE*, vol. 96, no. 9, pp. 1457–1486, 2008.

[62] J. Selvarathinam and A. Anpalagan, "Energy Harvesting From the Human Body for Biomedical Applications," *IEEE Potentials*, vol. 35, no. 6, pp. 6–12, 2016.

[63] V. Leonov, "Energy Harvesting for Self-Powered Wearable Devices," in *Wearable Monitoring Systems*, pp. 27–49, Boston, MA: Springer US, 2011.

[64] A. Khaligh, P. Zeng, and C. Zheng, "Kinetic Energy Harvesting Using Piezoelectric and Electromagnetic Technologies - State of the Art," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 3, pp. 850–860, 2010.

[65] F. Zhang, Y. Zhang, J. Silver, Y. Shakhsheer, M. Nagaraju, A. Klinefelter, J. Pandey, E. Carlson, A. Shrivastava, and B. Otis, "A Batteryless 19 uW MICS / ISM-Band Energy Harvesting Body Area Sensor Node SoC," vol. 48, no. 1, pp. 3–4, 2013.

[66] A. Khaligh, P. Zeng, X. Wu, and Y. Xu, "A hybrid energy scavenging topology for human-powered mobile electronics," *Proceedings - 34th Annual Conference of the IEEE Industrial Electronics Society, IECON 2008*, pp. 448–453, 2008.

[67] L. Shi, Z. Kabelac, D. Katabi, and D. Perreault, "Wireless Power Hotspot that Charges All of Your Devices," *MobiCom*, pp. 2–13, 2015.

[68] K. S. Moon, "Wireless Power Transferring and Charging for Implantable Medical Devices Based on Ultrasonic Resonance," no. July, pp. 12–16, 2015.

[69] C. T. Hager and S. F. Midkiff, "An analysis of Bluetooth security vulnerabilities," in *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 3, pp. 1825–1831, 2003.

[70] I. C. Society, *IEEE S tandard for Low- Rate Wireless Personal Area Networks (WPANs)*, vol. 2015. 2015.

[71] H. Li, Z. Jia, and X. Xue, "Application and analysis of ZigBee security services specification," *NSWCTC 2010 - The 2nd International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2, pp. 494–497, 2010.

[72] Y. Bin, "Study on Security of Wireless Sensor Network Based on ZigBee Standard," *Computational Intelligence and Security, 2009. CIS '09. International Conference on*, vol. 2, pp. 426–430, 2009.

[73] M. Toorani, "Security analysis of the IEEE 802.15.6 standard," *International Journal of Communication Systems*, 2016.