UNIVERSITY OF TARTU

FACULTY OF SCIENCE AND TECHNOLOGY

Institute of Computer Science

Computer Science Curriculum

Mari Liis Velner

Analyzing Predictive Features of Epileptic Seizures in Human Intracranial EEG Recordings

Bachelor's Thesis (9 ECTS)

Supervisor: Raul Vicente Zafra, PhD

Tartu 2017

Analyzing Predictive Features of Epileptic Seizures in Human Intracranial EEG Recordings

Abstract

Epilepsy seizure prediction is a challenge that scientists have tried to overcome throughout many decades, using different state-of-the-art features and machine learning methods. If a forecasting system could predict and warn epilepsy patients of impeding seizures in real time, it would greatly improve their quality of life. Seizure prediction consists of two stages: feature extraction from the data and sample classification to interictal (non-seizure) or preictal (pre-seizure) state. EEG data is commonly used, as it is inexpensive, portable and it most clearly reflects the changes in the brain's dynamics. While most studies focus on extracting novel features or using new classifiers, this Thesis focuses on ascertaining the most significant features among some that are commonly used in seizure prediction. Knowing which features influence the prediction results the most, helps to understand the inner workings of both the classifiers and the brain activity and to reduce the feature set in future research, making the classification process more effective. Intracranial EEG data of two patients was used in this Thesis with three classifiers from the scikit-learn library, which were combined with methods for evaluating feature importance. Moderately good to excellent prediction accuracies were achieved with these methods, which allowed to reliably analyze the feature importance results of the different classifiers.

Keywords

Epilepsy, EEG, seizure prediction, feature importance, machine learning

CERCS

P170, P176, B110

Inimeste intrakraniaalsete EEG salvestiste põhjal epileptiliste hoogude ennustamiseks sobivate tunnuste analüüs

Lühikokkuvõte

Epilepsiahooge on üritatud ennustada mitmeid aastakümneid, kasutades tipptasemel tunnuseid ja masinõppemeetodeid. Kui õnnestuks välja töötada süsteem, mis reaalajas hoiatab patsiente eelseisvate hoogude eest, parandaks see oluliselt patsientide elukvaliteeti. Epilepsiahoogude ennustamine koosneb kahest etapist: tunnuste ekstraheerimine ning näidiste klassifitseerimine hoogudevaheliseks (tavaline ajaperiood) või hooeelseks. Enamasti kasutatakse EEG andmeid, sest EEG on odav, transporditav ning väljendab muutusi ajudünaamikas kõige täpsemini. Kui enamik uuringuid keskendub uudsete tunnuste ekstraheerimisele või uute klassifitseerimisalgoritmide rakendamisele, siis antud bakalaureusetöö eesmärk oli välja selgitada, missugused kasutatavad tunnused on kõige olulisemad. Kui on teada, missugused tunnused kõige rohkem mõjutavad ennustamistulemusi, aitab see paremini aru saada nii klassifitseerimisalgoritmide tööprotsessist kui ka ajudünaamikast ning vähendada tunnuste hulka, mida masinõppes kasutada, muutes seega klassifitseerimisprotsessi efektiivsemaks. Bakalaureusetöös kasutati kahe patsiendi intrakraniaalseid EEG andmeid ning kolme algoritmi *scikit-learn* teegist, mida kombineeriti meetoditega, mis hindavad tunnuste mõju. Saadud ennustustäpsused olid mõõdukalt head kuni suurepärased ning võimaldasid seega analüüsida tunnuste mõju usaldusväärselt iga klassifitseerimisalgoritmi kohta.

Võtmesõnad

Epilepsia, EEG, hoogude ennustamine, tunnuste mõju, masinõpe

CERCS

P170, P176, B110

Table of Contents

1. Introduction

Epilepsy is the fourth most common neurological disease with only migraine, stroke, and Alzheimer's disease occurring more frequently [1]. It is characterized by recurrent seizures, which can be prevented by taking antiepileptic drugs, although patients respond to treatment only about 70% of the time. In many cases, seizures continue to occur even after the surgical removal of epilepsy-causing brain tissue [2] [3].

For epilepsy patients to live more normal lives, seizure forecasting systems are necessary. If EEG-based computational algorithms identify prior-seizure brain states early enough, a sufficient dose of medication could be administered or responsive electric stimulation applied to the brain in order to prevent seizures [3] [4].

EEG (electroencephalogram) signals, which are recorded from several electrodes placed inside the brain, are used to predict seizures. There are four distinguishable states in the epileptic brain: interictal (the normal state between seizures), preictal (prior to seizure), ictal (seizure) and post-ictal (after seizure). The main goal of seizure forecasting is to differentiate between the interictal and preictal states [3].

Seizure prediction consists of two stages: feature extraction from EEG data and sample classification. Feature extraction involves calculating the values of various statistical and analytical measures for every given EEG data sample to obtain information that is effective in differentiating interictal and preictal states. Much of previous research has focused on developing the best methods for the second stage, using state of the art features for predicting. This Thesis aims to analyze some of the more common features used for seizure prediction with EEG data and ascertain, which features have the most significant impact on the results.

Analyzing and understanding the significance of the features is important to ease biological interpretations of the computational solutions. If the influence of the features contributing to predicting is not understood, then having high accuracy in a machine learning pipeline is like a black-box with no insight to the reasons behind the feature set's success.

To analyze these features, the following stages were completed in this Thesis:

1. measure selection,
2. machine learning classifier selection,
3. feature extraction,
4. running the machine learning pipelines, and

5. feature importance analysis.

The American Epilepsy Society's intracranial EEG data sets of two human patients were used in this Thesis. Eighteen measures were selected based on their prevalence and performance in cutting-edge seizure prediction and their potential to differentiate between preictal and interictal recordings. By calculating these measures, features were extracted from the data using a moving window analysis and the resulting data was used in seizure prediction with three classifiers from the scikit-learn library: Gaussian Naïve Bayes Classifier, Random Forest Classifier, and the Logistic Regression Classifier. Feature importance was evaluated for every classifier using a separate, suitable technique and the results were compared and analyzed. In addition, the t-SNE algorithm was used to visualize the data.

The Thesis is organized as follows. Chapter 2 gives background information on epilepsy and seizure prediction. The data, measures and machine learning methods are described in Chapter 3. Results and data visualizations are presented in Chapter 4, limitations and possible future work are discussed in Chapter 5. Chapter 6 concludes the Thesis.

2. Background

This chapter gives an overview of the basic concepts of brain activity, epilepsy, EEG as a monitoring method, and machine learning and describes some of the previous works in seizure prediction.

The human **brain** consists of nerve cells i.e. **neurons.** The network, in which neurons transmit and gather electrochemical signals, contains millions of nerve fibers called dendrites and axons. This makes an overwhelmingly complex system, which is not easy to interpret [5].

Epilepsy is a disease, which involves recurrent seizures that are caused by abnormal excessive or synchronous neuronal activity within large groups of neurons. Epileptic seizures occur arbitrarily and can last for seconds or minutes [6]. Many causes of epilepsy exist, which vary by the age of the person, but unfortunately, the reason for the disease is unknown for about half of the cases [7].

There are multiple techniques for recording brain activity [8] of which **EEG** (electroencephalography) is the most commonly used with epilepsy patients. EEG uses electrodes, which are placed on the scalp or inside the brain, to detect the sum of electrical potentials of nearby neurons. The recorded brain activity (i.e. EEG signals) comprises of several distinct waves with different amplitudes and frequencies, which prevalence change in different states and processes such as during sleep, rest, wakefulness, and various pathologies. Some patterns are known to express a normal brain state, while deviations from this standard refer to an abnormal time period [6]. For instance, prominent beta brainwaves (about 12-30 Hz, cycles per second) indicate that the person is currently alert and engaged in a focused mental activity e.g. decision making [9].

Unlike the standard scalp EEG, **intracranial EEG** (iEEG) uses electrodes that are placed inside the brain. iEEG is invasive and thus, more risky for the patient, but it is closer to the seizure origin brain tissue and achieves better signal-to-noise qualities [10]. The electrodes' placement in the brain is decided based on clinical grounds. A single measurement from an electrode represents the total electrical potential produced by the neurons nearby the electrode. The number of measurements per second depends on the sampling frequency, which is usually 512 Hz or larger (i.e. 512 or more measurements per second) [6] [11].

Most of iEEG recordings unfortunately have an imprecise resolution concerning the source of the activity as parts of the brain far from the electrode can have a significant impact on the

recording. Nevertheless, iEEG provides invaluable information about the anatomical origin of the seizure's onset [11].

EEG is an important monitoring method for diagnosing epilepsy because it detects epileptic seizures as rhythmic signals that frequently coincide with or even precede the slightest changes in behavior. Therefore, EEG provides a possibility to avoid an impending seizure or to differentiate epileptic seizures from other diseases with convulsion-like symptoms [6].

Although seizures have an unpredictable nature, which makes seizure prediction a complicated task, there is strong evidence that seizures are processes that develop minutes to hours before the clinical onset [12]. There are four distinguishable states in an epileptic brain's dynamics [3]:

1. preictal (pre-seizure),
2. ictal (seizure),
3. postictal (after seizure), and
4. interictal (none of the above, normal brain state).

Seizure forecasting focuses on differentiating between **preictal** and **interictal** states. If a time period preceding a seizure can be correctly identified in real time via a device connected to the brain, it would give valuable time for patients to take appropriate action. Dangerous activities such as driving could be avoided, medications administered only when necessary to reduce side effects or electrical stimulation applied to reset brain dynamics [3] [13].

Seizure prediction can generally be divided into two steps. The first is **extracting features** from EEG records by calculating the values of different statistical and analytical measures. Feature extraction from EEG data is necessary as the raw dataset is too large to obtain prediction results in a reasonable time. In addition, signal processing extracts patterns that are effective in seizure detection, while the raw, noisy and seemingly random data would not yield any useful logic for the computer to successfully interpret [6].

The second step is **classifying** the samples with pre-calculated features into preictal or interictal states using statistical methods or machine learning algorithms [13].

Studies on seizure prediction using EEG recordings started in the 1960s [12]. Previous researches on seizure prediction have focused more on testing, developing or comparing different prediction algorithms or experimenting with new features; many of these notable researches are listed in [6]. This Thesis aims to compare the performance of eighteen univariate

measures, which are calculated using a moving window analysis on every electrode's recording. If a measure or feature is used to characterize a single electrode channel, then it is called univariate.

To illustrate some recent results in seizure prediction, a few additional studies are described below.

The Kaggle 2014 seizure prediction challenge's [14] winning team used a weighted average of three different models, including Random Forest Classifier [15]. In addition to the two patients' datasets used in this Thesis for feature extraction, they also included iEEG data of five dogs. A 72% percent accuracy was achieved on the test data with the Random Forest model, for which 80 trees and 8 second time windows with overlap were chosen as parameters. A description of the Random Forest classifier model can be found in chapter 3.4.2 and the time window analysis is further explained in the Feature Extraction chapter.

A recent research that only used the data of the two patients, was conducted by Kumar et al. in 2015 [4]. The 10 minute data segments were split into overlapping 10 second windows using moving window analysis and over 10 measures were used in feature extraction, including mean spectral power from several frequency bands and kurtosis, also used in this Thesis. Among the four classifiers, Random Forest outperformed all of them, including the much more commonly used Support Vector Machines. They concluded that windowing long data segments improves the classification performance.

The data, extracted features and machine learning methods used in this Thesis are described in the following chapter on methodology.

The code for this Thesis can be found here: https://github.com/mariliisvelner/epilepsy-seizure-analysis.

3. Methodology

The following subchapters present detailed descriptions of the iEEG datasets used in this Thesis, the feature extraction process, and the different measures that were calculated as features. In addition, the three classification algorithms, feature importance analysis methods and the t-SNE algorithm are also described in these subchapters.

3.1 Datasets

iEEG datasets of Patient 1 and Patient 2 from the 2014 Kaggle American Epilepsy Society Seizure Prediction Challenge are used in this Thesis for feature extraction. Both datasets consist of 10 minute iEEG data clips (segments) of a human patient with epilepsy. These data segments are labelled as "interictal" for non-seizure data segments and "preictal" for pre-seizure data segments. The datasets also contain unlabeled "test" segments, which were not used in this Thesis [14].

Each data segment is stored in a .mat file (a MATLAB format) that contains a matrix of iEEG recorded values arranged with rows as electrodes and columns as time points. In other words, every row is a recording from a single electrode. The iEEG data for Patient 1 was recorded using 15 electrodes and for Patient 2 using 24 electrodes. The sampling rate for both datasets was 5000 Hz (5000 data measurements per second) [14].

For Patient 1, there were 50 interictal segments and 18 preictal segments in the original dataset, which makes 500 minutes of interictal data and 180 minutes of preictal data. The dataset of Patient 2 consisted of 42 interictal segments (420 minutes) and 18 preictal segments (180 minutes). Thus, there is a total of 680 minutes of iEEG data for Patient 1 and 600 minutes of data for Patient 2.

Both preictal and interictal data segments in these datasets form one hour sequences. Preictal data segments have at least a five minute seizure horizon (i.e. from 1:05 to 0:05 before seizure onset) to prevent seizure signals from appearing in the segments. The interictal segments are separated from seizures by at least four hours to avoid contamination with preictal or postictal signals [14].

Below are figures to illustrate the data used in this Thesis. Every line corresponds to a single electrode's recording, the first being from electrode number 1 and the lowest from the 15th electrode. Figure 1 shows a 5 minute section of interictal iEEG data from Patient 1 and Figure 2 illustrates 5 minutes of preictal data. For a more detailed view, the first second of the interictal

data depicted in Figure 1 is shown in Figure 3 and the first second of preictal data in Figure 2 is presented in Figure 4.

These figures were obtained using the code in the file *eeg_visualization.py*.

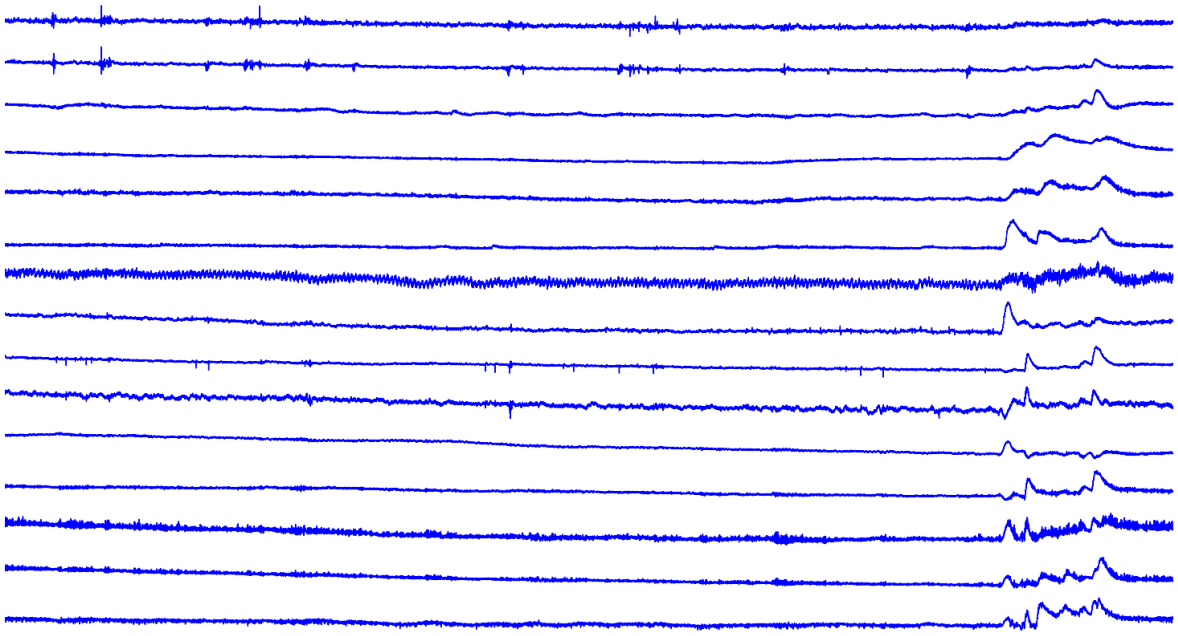

Figure 1. Five minutes of interictal data from Patient 1.

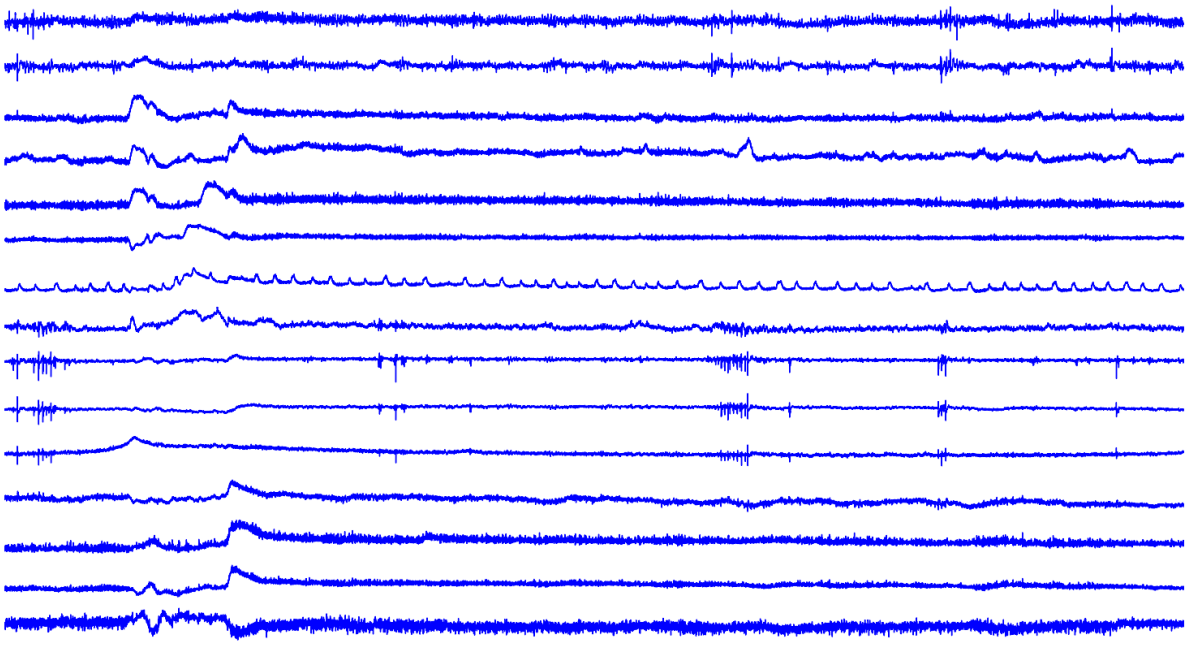

Figure 2. Five minutes of preictal data from Patient 1.

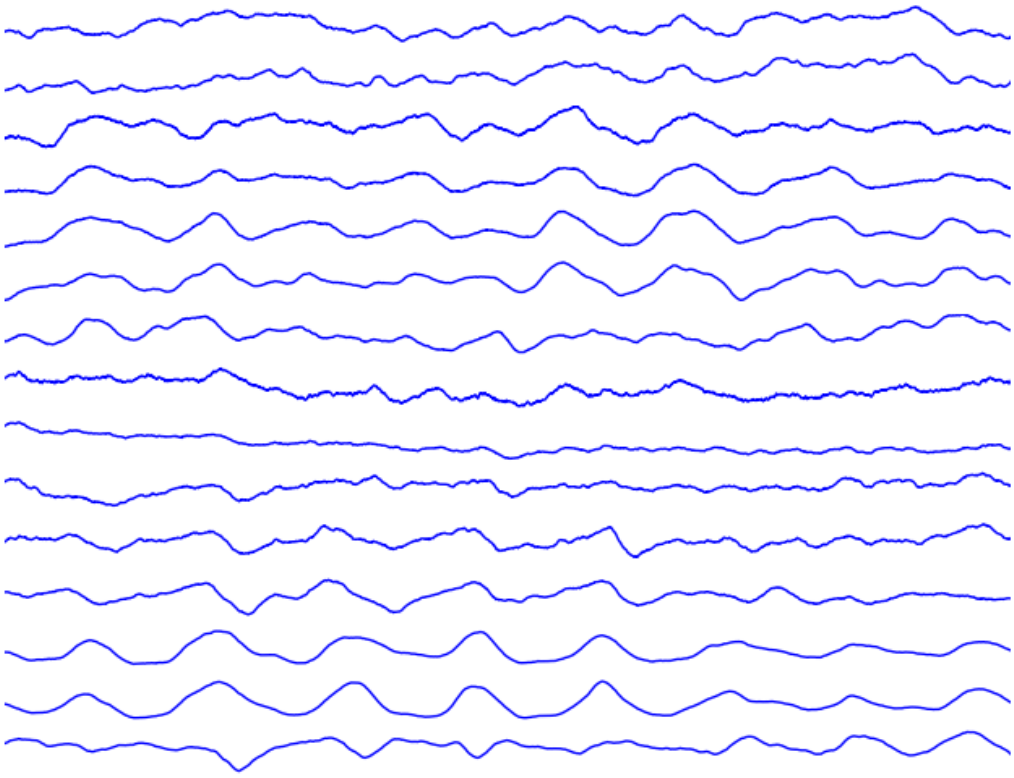

Figure 3. One second of interictal data from Patient 1. This is the first second of the interictal data shown in Figure 1.

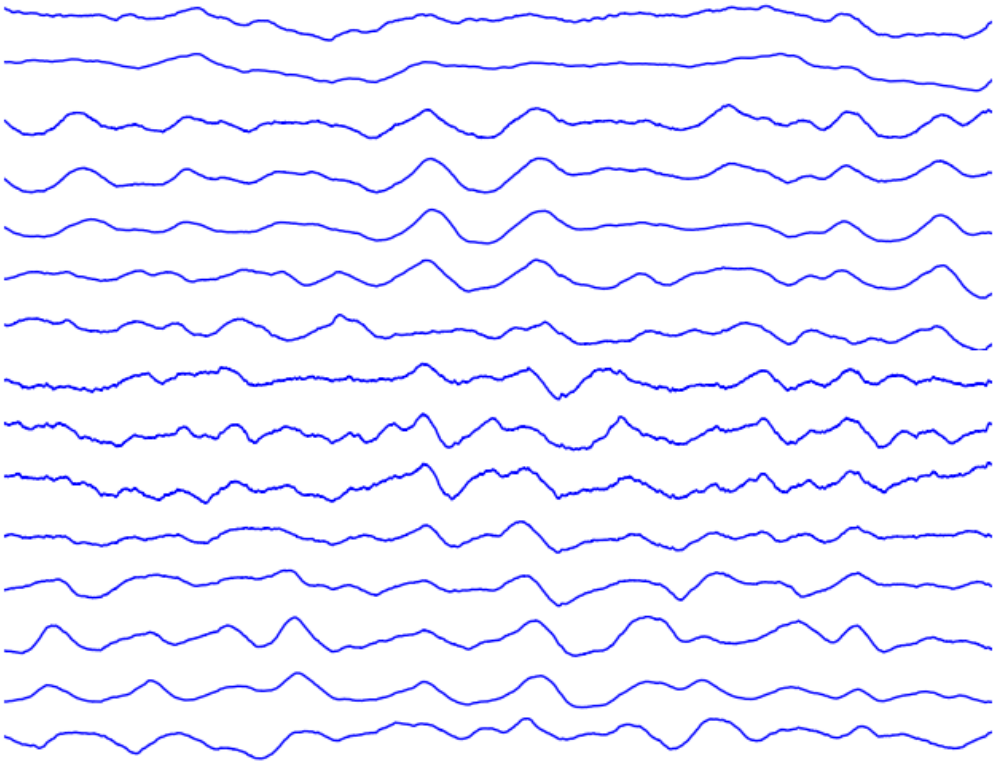

Figure 4. One second of preictal data from Patient 1. This is the first second of the preictal data illustrated in Figure 2.

As seen from these figures, it is not trivial to detect whether the current time period is preictal or not. Abnormal signals might occasionally appear in the interictal state (for instance, the last minute of interictal data in Figure 1), indicating some kind of physical or mental activity. Some electrode channel might also show different behavior than the others (e.g. electrode number 7 in Figure 1), but not the preictal kind. Thus, it is important to find the most effective measures to describe the differences between interictal and preictal iEEG data.

3.2 Feature Extraction

The process of feature extraction consisted of three main steps:

1. dividing electrode recordings into smaller time windows;
2. calculating features for every time window;
3. concatenating features of concurrent time windows to one sample.

These steps are described in detail below.

Two datasets were calculated for both patients using a non-overlapping window technique. Specifically, for each data segment, every 10 minute data row (corresponding to a single electrode) in the matrix was divided into smaller time windows. This means that for every 10 minute data segment, each electrode yielded $10 * \frac{60}{w}$ time windows, where w is the length of the time window in seconds. Thus, the entire dataset produced a total number of $s * x * 10 * \frac{60}{w}$ time windows, where s is the number of data segments and x is the number of electrodes.

For each time window, 18 features were calculated to represent the 18 measures. Then, the features of the time windows occurring at the same time were concatenated to form a single sample, a data row in the new dataset. Therefore, each sample in the final dataset represented a certain time period and had a total number of $18 * x$ features, where x is the number of electrodes. Every feature name was denoted with a number, which marked the electrode's number where the window originated from. For instance, if the measure p was calculated from a time window that belonged to a recording from electrode number 1, then the feature was named "$p1$". Every sample also had a class: 1 for interictal or 2 for preictal, which were the target values in predicting.

This procedure yielded a new dataset with $18 * x$ columns (in addition to the class) and $s * \ 10 * \frac{60}{w}$ rows, where x is the number of electrodes and s is the number of data segments. Each row in this dataset represents a time window. Following this procedure, datasets with 10

second windows and 60 second windows were calculated for both patients, making four datasets in total.

Using moving window analysis, where linear measures are calculated from a window of EEG data with a certain length, is a common practice in seizure prediction. The duration of the time window, usually from 10 to 40 seconds, is important, as smaller windows hold too much detail and larger windows are too abstract [4] [12].

The process for a single segment is illustrated by Figure 5.

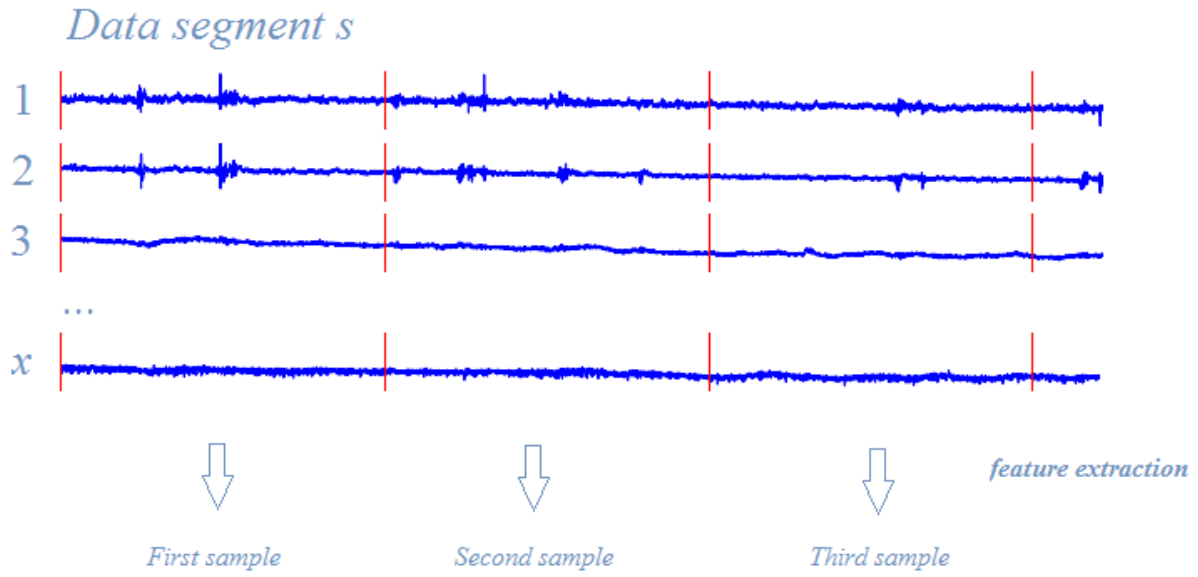

Figure 5. Feature extraction process for a single data segment s. 18 measures are calculated for every time window of every electrode (1 … x) and the features of concurrent windows are concatenated to form new samples.

Table 1 describes the new datasets that were calculated from the Kaggle datasets.

Table 1. Details of the new datasets.

Dataset	Patient	Window length (s)	Electrodes	Features	Windows
1	1	10	15	18 × 15 = 270	680 × 6 = 4080
2	1	60	15	18 × 15 = 270	680 × 1 = 680
3	2	10	24	18 × 24 = 432	600 × 6 = 3600
4	2	60	24	18 × 24 = 432	600 × 1 = 600

The number of features in the new dataset is calculated by multiplying the number of measures with the number of electrodes in the original dataset. In order to get the number of windows

(rows), the original dataset's length in minutes must be multiplied with either 1 for the minute-window datasets or with 6 in case of 10 second window datasets.

The new datasets with 60 second windows and 10 second windows are hereinafter referred to as *60 second dataset* and *10 second dataset* respectively.

The 18 measures extracted from the raw data are the following:

- Hjorth activity, mobility, and complexity,
- Higuchi fractal dimension,
- skewness,
- kurtosis,
- spectral power for delta, theta, alpha, beta, low gamma and high gamma frequency bands, and
- spectral power in each of the previously mentioned frequency bands normalized by total power in all of the frequency bands.

These measures are further described in the following subchapters.

To understand the essence of these features, some concepts are explained as follows. When a signal is said to be in the **time domain**, it means that each value in the signal is an amplitude, which corresponds to a point in time. Time domain parameters are parameters that are calculated from data (in this case, iEEG signals) in the time domain.

The **Fourier Transform** (FT) is a technique that takes a signal in the time domain and converts it to the **frequency domain**. This representation of the signal is called **the power spectrum**. Power spectrum essentially measures how much of a sinusoid with a certain frequency does the signal contain. FT decomposes a signal into frequencies that it consists of and finds the amplitudes, weights of these oscillations. The power spectrum is therefore a representation of the signal, where the x-axis represents frequencies of different sinusoids and the y-axis the weights of those frequencies. A large weight indicates that the sinusoid with the corresponding frequency is one of the main components of this signal [16]. These weights are also referred to as **spectral power**.

The code for feature extraction can be found in the file *make_data.py*.

3.2.1 Hjorth Parameters

Hjorth parameters – activity, mobility, and complexity – are time domain measures that are very useful in analyzing EEG data [17] and are commonly used in seizure prediction [12]. The parameter **activity** represents the variance of the signal's amplitude, that is, it measures how far the amplitude deviates from the mean amplitude:

$$Activity = var(f(t))$$

where $f(t)$ represents the signal and var the variance function [18]. The variance of the signal has previously exhibited a decrease during preictal time [19].

Mobility is the square root of the variance of the signal's first derivative divided by the activity of the signal [18]:

$$Mobility = \sqrt{\frac{var\left(\frac{df(t)}{dt}\right)}{var(f(t))}}$$

Complexity represents the ratio of the mobility of the signal's first derivative divided by the mobility of the signal [18]:

$$Complexity = \frac{Mobility\left(\frac{df(t)}{dt}\right)}{Mobility(f(t))}$$

According to Kaboli et al. [20] these parameters can be described in the frequency domain, despite the fact that they are defined in the time domain. Specifically, activity represents the total power of the signal, mobility is an estimate of the mean frequency and complexity describes how similar is the shape of the signal to a pure sine wave.

The Hjorth parameters were calculated using the pyEEG library for Python [21].

3.2.2 Higuchi Fractal Dimension

Fractal dimension is a time domain measure which describes the complexity of a time series. That is, if the fractal dimension increases, so does the degree of complexity [18]. This feature has been used to detect specific states of physiologic function in EEG analysis [22]. Higuchi's algorithm is one of the various algorithms that have been developed to calculate the fractal dimension and is one of the more accurate estimations of a signal's fractal dimension. The construction of the algorithm is analyzed in [22].

The Higuchi fractal dimension feature was calculated using the pyEEG library for Python [21].

3.2.3 Skewness and Kurtosis

Skewness and kurtosis are statistical measures of a time series.

Skewness is positive, if the mode is less than the median, which means that small values dominate over the large values in the time series. In case of a positive skewness, it is said that the distribution graph is skewed to the right [23].

Skewness is negative if the mean value is less than the median, which indicates that the number of large values in the time series is bigger than that of small values. In this case, the distribution graph is skewed to the left [23].

In a time series, where the distribution of values is even, the skewness value is zero.

Figure 6 illustrates the shape of the distribution graph with different skewness values.

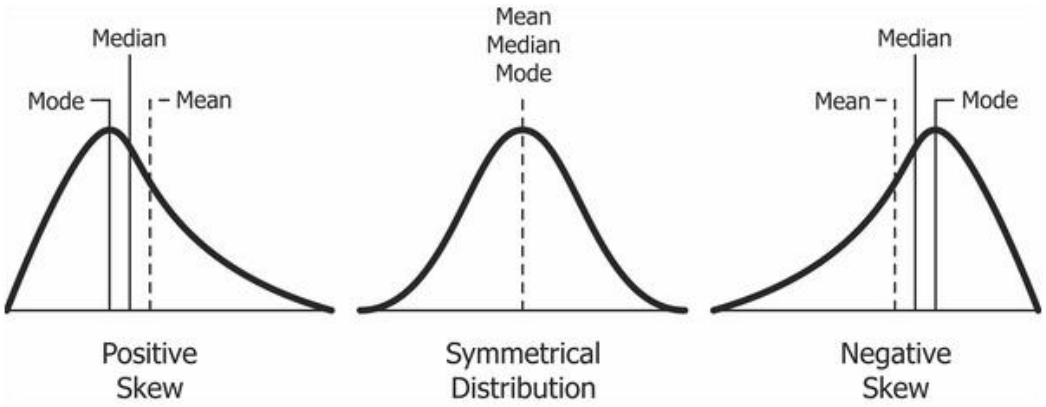

Figure 6. Skewness in different frequency distribution graphs [24].

Kurtosis is low when most of the values in the time series are clustered around the mean. If intermittent extreme values dominate, then the time series has a high kurtosis value [25]. Kurtosis has been shown to increase during preictal time in several studies [26].

The skewness and kurtosis features for the time windows were calculated using the SciPy library's statistical functions *skew* and *kurtosis* respectively.

3.2.4 Spectral Band Power and Band Power Ratio

Brainwaves are a product of synchronous electrical pulses from masses of neurons that communicate with each other. They are measured in Hertz (cycles per second) and are divided into bandwidths or bands in order to categorize them by their functions [9].

The main frequency bands in brain activity are the delta, theta, alpha, beta and gamma bandwidths. These bandwidths have a biological significance in human brain activity [9] and are often used in seizure prediction. The specific division of these bands may vary by some hertz or certain bands (e.g. gamma) may be divided into several bands (low and high gamma) but the general concept and essence of these measures remains the same.

Spectral power features have been often and successfully used in seizure prediction [12] [26]. In this Thesis, spectral band power is calculated for the frequency bands delta (0-4 Hz), theta (4-8 Hz), alpha (8-12 Hz), beta (12-30 Hz), low gamma (30-70 Hz) and high gamma (70-180 Hz). In other words, for each band, the spectral power feature is calculated by summing up the spectral powers of the frequencies in these bands, thus yielding six features for an electrode in a single time window.

The spectral power ratio feature of each band is calculated by dividing the band's spectral power with the total power in all of the frequency bandwidths.

Both of these features were calculated using the pyEEG module for Python [21].

3.2.5 Notation

This subchapter presents the notation used for the measures and features in the text.

The notation of the **measures** is presented in Table 2.

Table 2. Notation of measures.

Measure	Notation
Hjorth activity	activity
Hjorth mobility	mobility
Hjorth complexity	complexity
Higuchi fractal dimension	hfd
Skewness	skewness
Kurtosis	kurtosis
Spectral power of the delta band	ps_delta
Spectral power of the theta band	ps_theta
Spectral power of the alpha band	ps_alpha
Spectral power of the beta band	ps_beta
Spectral power of the low gamma band	ps_lowgamma
Spectral power of the high gamma band	ps_highgamma

Spectral power ratio of the delta band	psr_delta
Spectral power ratio of the theta band	psr_theta
Spectral power ratio of the alpha band	psr_alpha
Spectral power ratio of the beta band	psr_beta
Spectral power ratio of the low gamma band	psr_lowgamma
Spectral power ratio of the high gamma band	psr_highgamma

The **feature** notation is as follows. A number, which denotes the electrode, from which the feature value originates, is added to the corresponding measure's name. For example, "skewness08" refers to the skewness of the iEEG data window recorded by the 8th electrode. To avoid confusion, the individual attributes, features without the electrode numbers will be hereafter referred to as **measures**. For instance, *hfd08* is a feature – it is used in the classification processes, while *hfd* is a measure, which represents the Higuchi fractal dimension.

3.3 Visualization by t-SNE

t-distributed stochastic neighbor embedding (**t-SNE**) is an algorithm for visualizing high-dimensional data (i.e. data with many features). It gives each sample a location in a two- or three-dimensional map, in which clustered points represent similar data samples, while large distances between points reflect the large differences between the corresponding samples. t-SNE is able to capture a lot of the local structure of the high-dimensional data excellently and also reveal larger structures like clusters at several scales [27].

Data visualization is important as it reveals the structure of the complex dataset in a way that the seemingly arbitrary feature values do not. This might allow to make some parallels with the prediction results, as data which does not yield any concrete structure or has data points of different classes intermingled is harder to categorize.

The *TSNE* algorithm of the scikit-learn toolkit was used in this Thesis to visualize the data in combination with the *TruncatedSVD* method. The default values for the parameters were used and the data was reduced to two dimensions.

The visualizations are presented in chapter 4.1.

3.4 Machine Learning Methods

The code for this Thesis was written in Python and the following three classifiers of the scikit-learn library were used for machine learning: Random Forest Classifier, Logistic Regression

Classifier and Gaussian Naïve Bayes Classifier. Other algorithms (Support Vector Machines, Stochastic Gradient Descent, and Multi-layer Perceptron) were also tested but were found unfit for this Thesis based on their low accuracy or slow performance, which would have hindered the work process too much. In addition to successful seizure prediction, high accuracy is necessary for obtaining reliable results in analyzing feature importance.

The scikit-learn library was chosen for this Thesis for the following reasons [28]:

- the author was already familiar with it,
- it implements all of the three machine learning algorithms used in this Thesis,
- it is widely used in scientific computing,
- it is open-source,
- it is easy to use,
- it has few dependencies, and
- has proper documentation.

The following subchapters will describe each of the three machine learning methods.

3.4.1 Gaussian Naïve Bayes Classifier

In case of real-valued, continuous data, it is typically assumed that the continuous values associated with the classes follow a normal (Gaussian) distribution [29]. The data used in this Thesis has real-valued features.

The Gaussian Naïve Bayes classifier (GNB) implements the Gaussian Naïve Bayes algorithm, which assumes the probability of the features to be Gaussian [30]:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{\left(x_i - \mu_y\right)^2}{2\sigma_y^2}\right)$$

The following parameters and variables are used in this equation:

- x_i (the feature x with the value i),
- y (a class),
- σ_y^2 (variance of the feature x's values in class y; measures, how far values are spread out from their mean),
- μ_y (the mean of the attribute x's values in class y).

In other words, the algorithm computes the probability of x_i being of class y by inserting the value x_i to a Gaussian distribution (a.k.a. Normal distribution) equation along with the parameters σ_y^2 and μ_y.

This classifier is used for the purpose of being the base, naïve method to evaluate the other more complex methods' performances by comparing their prediction results to GNB's results.

The class *GaussianNB* of the scikit-learn library is used in this Thesis as the Gaussian Naïve Bayes classifier. All of the parameters were kept to their default values and are further described in [31].

3.4.2 Random Forest Classifier

Random forest classifier (RFC) or random decision forest uses a set of decision trees to classify a sample.

A **decision tree** is basically a series of if-then statements and when a sample in the dataset is applied to it, the sample is classified. Therefore, if all the samples in the dataset are run through the decision tree classifier, each individual sample is classified [32].

Figure 7 is an example of a simple decision tree, which is applied to a dataset, where each sample represents a day and has the attributes "outlook", "humidity" and "wind". Using this tree, every record in the dataset is classified as being of one class or not (e.g. is it an adequate day to play golf or not) [33].

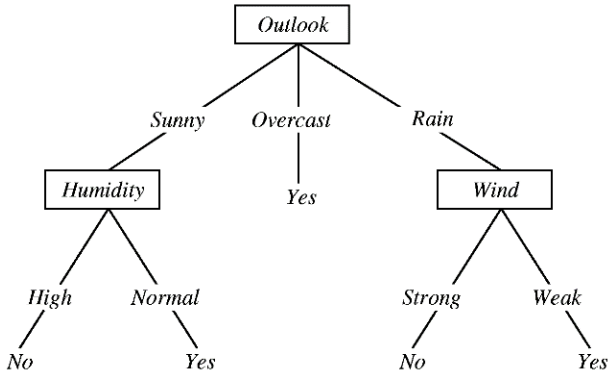

Figure 7. Decision tree example [33]. To classify a sample from the dataset, the decision making process is started from the root of the tree and moved to the bottom, choosing the branches according to the values of the attributes. If a leaf node is reached, then the sample is classified [32]. For instance, if a sample with the outlook of "sunny", "normal" humidity and wind with the value "strong" is applied to the decision tree in Figure 7, the classification process starts from the root node and selects the correct branch corresponding to the outlook value. In this case, the leftmost branch is selected, after which the process moves to the next node, which compares the humidity value. As the sample's humidity is "normal", then the right

branch is selected, which directs to a leaf node. The leaf node's value is "Yes" and thus, the sample is classified as being a good day to play golf.

Decision trees have high execution speed but fail to grow in complexity while still maintaining high generalization accuracy. This limitation means suboptimal accuracy on training data, and RFC attempts to mitigate this flaw by constructing several trees whose efficiency can be arbitrarily increased with increasing accuracy for both training and new data [34].

RFC randomly selects a number of different subsets from the features and builds a decision tree in each of those using the entire training set. This way, the decision trees generalize their classification in different ways and avoid extensive overfitting on the training set. The trees in different subspaces of features are all used to classify a sample. The output class of the forest is calculated by using a discriminant function on the classifications of the decision trees or by choosing the mode from the trees' classifications [34].

The class *RandomForestClassifier* of the scikit-learn library is used in this Thesis as the random decision forest algorithm. The number of trees in the forest was left to its default value of 10. The number of features to consider when making subsets of the feature space was set to the total number of features, to obtain results, which encompass all of the features. The mode of applying weights to the classes was set to "balanced" in order to take into account the frequencies of the two classes. All of the other parameter values were left to their default values to avoid overfitting and can be found in [35].

RFC was used in this Thesis because it is an extremely fast classifier that even with a large feature set is very efficient in its classification process. Furthermore, the method has been used in seizure prediction on the same dataset with successful results by Tieng et al. [15] in Kaggle's American Epilepsy Society Seizure Prediction Challenge and Kumar et al. [4]. RFC can also automatically model the interactions between features, which makes it easier to analyze the features' influence on the result [36].

3.4.3 Logistic Regression Classifier

Logistic regression classifier (LRC) is an algorithm that assigns one of two classes to a data record by calculating the probability of the sample belonging to one of the classes. LRC uses a **logistic function**, which maps a real-valued number into a value between 0 and 1 to classify samples [36].

LRC works by extracting a set of weighted features from the sample, combining them linearly and applying the logistic function to this combination. The result of the equation is a value

between 0 and 1 and the class is determined by whether this value is less than 0.5 or not [36] [37].

The feature coefficients are learned from the training data by using **maximum-likelihood estimation**, which essentially chooses the weights that make the classes of the training samples more probable [36].

The class *LogisticRegressionCV* of the scikit-learn library is used in this Thesis to perform logistic regression classification on the data. The "liblinear" algorithm was used as the *solver* parameter for this method [38]. This algorithm is used as the optimizing function for maximum-likelihood estimation of the *LogisticRegressionCV*. The value "l1", which means L1 regularization, for the parameter *penalty* was found to be the most efficient. A **penalization** or **regularization** algorithm is used to avoid overfitting. If a feature helps to predict the outcome perfectly only because it occurs in one single class, it will be given a very high weight and thus, some weights will try to strictly fit the patterns of the training data and therefore will fail to generalize on new data. L1 regularization is the sum of the absolute values of the weights. This penalty with its own weight is subtracted from all of the coefficients to reduce overfitting [36].

All of the other parameter values were left to their default values to avoid overfitting and can be found in [38].

LRC was used in this Thesis due to its ability to work well on large datasets and reduce overfitting by using penalties [36].

3.5 Feature Analysis

In parallel with running the machine learning pipelines, feature importance analysis was also performed using different methods for each of the three classifiers. These methods are described in the following subchapters.

3.5.1 SelectKBest for Naïve Bayes

The class *SelectKBest* from scikit-learn was chosen to ascertain, which features most influence the prediction result. This method selects features according to the *k* highest scores that are calculated with the *f_classif* function. This function computes the one-way ANOVA (analysis of variance) F-values for every sample. **One-way analysis of variance** essentially tests if a numeric variable (values of some feature) differs according to a categorical variable's values (values of the class) [39].

3.5.2 Feature Importance for Random Forest

The feature importances for RFC were acquired from the scikit-learn *RandomForestClassifier* model's attribute *feature_importances_*, which (after training the model) returns the feature importances from the fitted model [35]. These importances are implemented using **mean decrease impurity** (MDI), which is calculated as follows. As known, a random forest consists of several decision trees, where each non-leaf node is a condition based on a single feature. This condition splits the dataset into several subsets and the feature based on which the decision is made, is called impurity. When training a tree, the number of times a feature is chosen as the impurity can be counted and divided with the number of samples it splits. This ratio is the MDI [40].

3.5.3 Coefficients for Logistic Regression

In case of LRC, feature importance was computed as the absolute value of the feature's coefficient. This expresses how much impact the feature has on the prediction result.

4. Results

This chapter presents the results in three subchapters. Firstly, data visualizations using the t-SNE algorithm are given in Chapter 4.1. Chapter 4.2 describes the results of the classification processes and feature importance evaluation for every classifier. The final subchapter summarizes the previously described results.

4.1 t-SNE Visualization

This chapter presents the t-SNE visualizations of the data. Visualization is important to get an insight to the general structure of the data: which clusters appear, how many clusters there are, do samples of different classes appear in separate groups or are mixed up and thus, hard to categorize etc.

Data was visualized using a combination of the *TruncatedSVD* and *TSNE* methods of the scikit-learn library.

Figure 8 depicts the t-SNE visualizations of all of the four datasets. The upper plots are visualizations of the Patient 1 datasets: the left panel represents the 60 second dataset and the right panel illustrates the 10 second dataset. The lower plots depict the second patient's datasets: the 60 second dataset is presented on the left and the 10 second dataset on the right.

Interictal windows are represented by the lighter, pink data points, while the preictal windows are depicted as the green data points.

The coordinates of these data points do not represent their values and are just for the purpose of structuring the data. The data point's location in relation to other data points is important as it gives a perspective of the distance, difference between the samples they represent.

In Figure 8 we can observe that the windows of different classes are quite intermingled and do not form groups that are easily distinguishable. This is more of a case for the 60 second datasets, where very few clusters of a single class appear. For the Patient 1 60 second dataset, about four clusters of interictal data points can be distinguished (two on the bottom of the plot, one on the right and one on the upper left). A single cluster of preictal data points appears on the bottom left corner. However, for Patient 2 no such clusters consisting of data points of a single class can be distinguished. This implies that the second patient's preictal and interictal data is more difficult to differentiate and thus, less accurate prediction results might be expected.

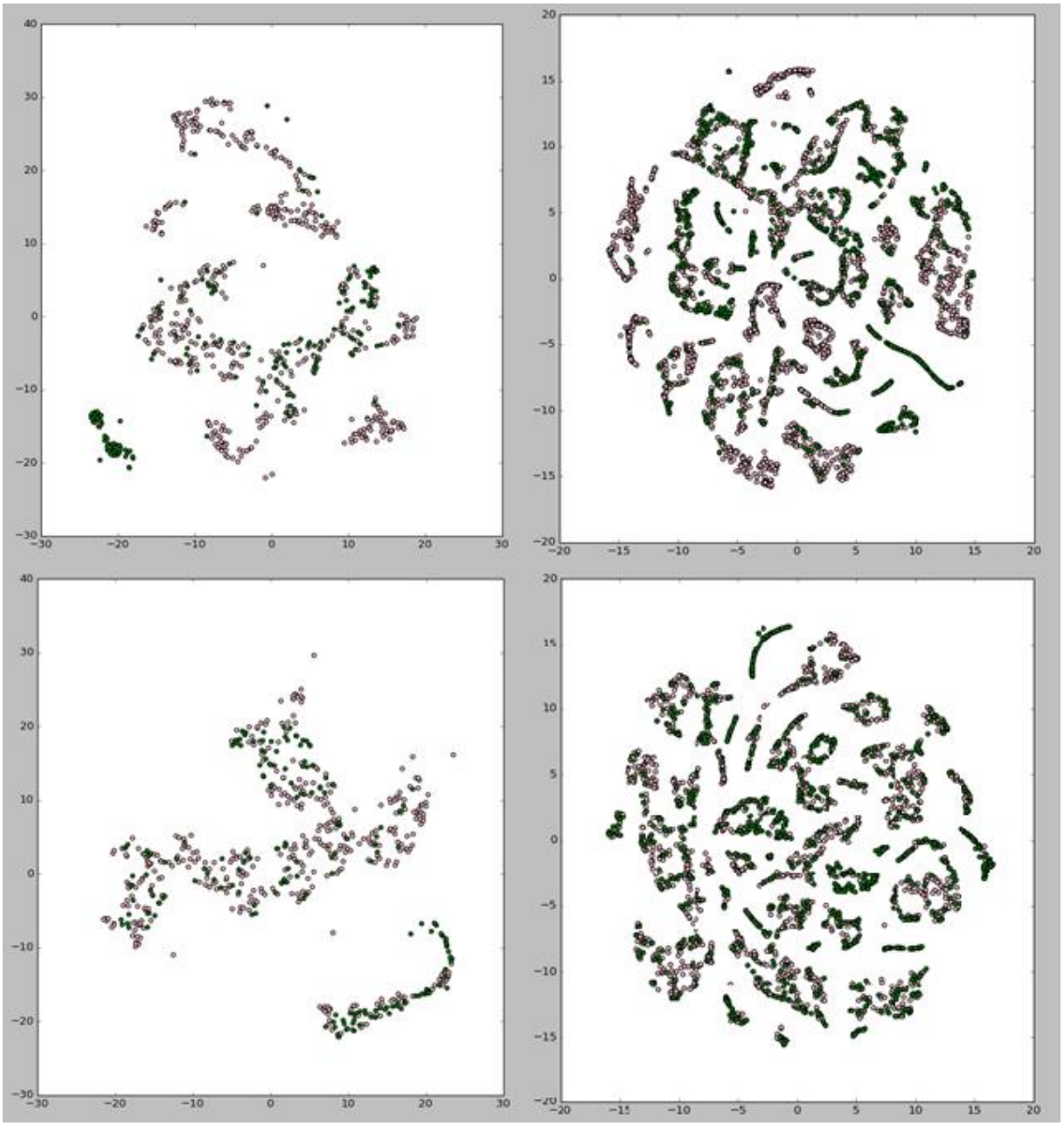

Figure 8. The t-SNE visualization of the four datasets. The upper left plot represents the Patient 1 60 second dataset, the upper right plot the first patient's 10 second dataset. Lower plots depict the second patient's 60 second and 10 second datasets on the left and right respectively. The pink data points represent interictal windows of the dataset, while the green data points represent preictal windows.

As there are six times more windows in the 10 second datasets, a lot more clusters are likely to appear. This assumption is confirmed by the two plots in Figure 8. There are a significant amount of preictal and interictal clusters for both 10 second datasets and overall less large clusters appear, in which data points of different classes are distributed evenly.

The code for generating these plots can be found in the file *tsne.py.*

4.2 Machine Learning Results

This chapter presents the results acquired in the classification processes. The first subchapter describes data partitioning to the test and training set, and the following three subchapters present the prediction accuracies and analyze the feature importances for each classifier.

4.2.1 Data Partitioning

To create better conditions, ease the interpretation of the results and to mitigate the shortage of preictal data, the training and test data contained the same number of preictal and interictal windows. These windows were selected randomly from the whole data in two ways:

1. **Window-based partitioning (W)**. In this case, preictal data was randomly split in half between the training and test set, and then the same number of randomly selected interictal windows were added to both sets.
2. **Segment-based partitioning (S)**. For this case, preictal **segment numbers** were divided randomly in half between test and training set. Then, **windows with those segment numbers** were added to the corresponding sets. The same procedure was repeated with interictal segment numbers and windows, selecting **the same number of segments (windows)** as preictal segments (windows).

Specifically, for the second partitioning, half of the preictal segment numbers were randomly chosen and the windows of those segments were included in the training set, while the other half was added to the test set. An equal number of segments (as there were preictal segments) was randomly chosen from the interictal segment numbers and then randomly divided in half between the training and test set. This ensured that the training and test set did not contain windows of the same segments and thus, the data was more likely less correlated.

These different splitting methods will be hereinafter referred to as **partition (partitioning) W** and **partition (partitioning) S** respectively.

Classification processes for every dataset and every method were executed with both of these partitions. This was done to observe, how much the correlation of partition W affects the prediction results compared to the less correlated partition S.

Table 3 illustrates the training and test sets for both of the two partitions above.

Table 3. Training and test sets by segment numbers.

Patient	Preictal segments	Interictal segments	Preictal segments in training set	Preictal segments in test set	Interictal segments in training set	Interictal segments in test set
1	18	50	9	9	9	9
2	18	42	9	9	9	9

This equal selection of interictal and preictal windows was chosen due to the large imbalance between the number of interictal and preictal segments in the original dataset, which resulted in a very low prediction accuracy that does not allow to make reliable comparisons between feature importances. In addition, splitting the data in the described way shortened the execution time of the program, which in turn helped to avoid hindrances in the work process.

To compare the results of three classifiers, cross-validation was performed with each of them, using 5 folds and splitting the data between the training and test set using partitions W and S. The feature importance scores presented in the following chapters are the average scores over all of the folds.

The following chapters will present the results of predictions and feature importances by comparing an algorithm's performance on all of the datasets. As a reminder, the notation described in chapter 3.2.5 will be used to denote measures and features.

4.2.2 Gaussian Naïve Bayes Classifier

This chapter will present the classification accuracies and feature importance analysis for the Gaussian Naïve Bayes classifier.

For Gaussian Naïve Bayes classifier (GNB) the classification was performed in two ways. Firstly, GNB was used in combination with the *SelectKBest* (SKB) method. SKB first selected the data of the ten best features from the entire dataset and the new dataset was then applied to the GNB.

Secondly, to get an estimate of SKB's impact on the prediction results, the whole data was fed to GNB without using SKB.

The following two tables present GNB's prediction accuracies and feature importance results, which were acquired using partitioning W in cross-validation.

Table 4 shows GNB's prediction accuracies for partition W.

Table 4. Partition W prediction accuracies for GNB.

Patient	Window length (s)	Prediction accuracy without SKB (%)	Prediction accuracy with SKB (%)
1	60	65	80
1	10	64	77
2	60	55	78
2	10	53	78

In Table 4 we can see that using the SKB method improves the prediction result by 13-25%. While for Patient 1 GNB without SKB shows some capability of producing moderate results, the accuracies for Patient 2 are almost at 50%, making the classifier as ineffective as one that would predict every sample to be of one class. With SKB, the accuracy is moderately precise as would be expected of a statistical method on complex data.

Table 5 shows the ten features selected by SKB. The values in the parentheses indicate the mean values of the scores assigned by SKB to the corresponding features in every fold. The last row represents the prediction accuracies also presented in Table 4.

Table 5. The selected 10 features and prediction accuracies using SKB for partition W.

	Patient 1 (60 s)	Patient 1 (10 s)	Patient 2 (60 s)	Patient 2 (10 s)
1.	psr_alpha07 (229.8)	psr_theta15 (1080.5)	hfd08 (226.4)	hfd08 (1239.9)
2.	psr_theta06 (211.9)	psr_theta06 (1051.6)	hfd06 (172.6)	hfd06 (972.1)
3.	psr_theta15 (202.1)	psr_theta14 (1005.1)	hfd10 (163.5)	hfd13 (949.9)
4.	psr_theta14 (200.3)	psr_alpha07 (941.7)	hfd13 (160.5)	hfd05 (896.8)
5.	psr_theta13 (173.1)	psr_theta13 (860.1)	hfd16 (148.6)	hfd16 (896.6)
6.	psr_theta05 (146.0)	psr_theta07 (693.2)	hfd24 (146.6)	hfd24 (892.5)
7.	psr_theta07 (137.4)	psr_theta05 (672.8)	hfd05 (141.1)	hfd10 (871.0)
8.	psr_alpha06 (128.5)	psr_alpha06 (552.3)	hfd17 (132.3)	hfd17 (798.2)
9.	psr_theta12 (102.4)	psr_theta12 (483.6)	hfd15 (129.9)	hfd15 (792.5)
10.	psr_alpha12 (97.9)	psr_alpha12 (470.8)	hfd07 (124.2)	hfd23 (741.1)
%	**80**	**77**	**78**	**78**

These results imply that the prediction process is very similar for datasets of the same patient as the selected ten features for Patient 1 are exactly the same for the 10 second and 60 second datasets, only with slightly different ranking. This applies also for Patient 2, which has only one feature per list that does not appear in the other one (*hfd07* for the 60 second list and *hfd23* for the 10 second list). While *psr_theta*, and therefore also the theta band are dominant among the measures of Patient 1, *psr_alpha* also appears several times. However, there is not a single electrode that can be stated to be the most important, as electrodes 7, 12 and 15 all appear twice for both 10 second and 60 second datasets.

While the spectral features dominated the ranking for patient 1, fractal dimension is the only measure selected for both datasets of Patient 2. This does not allow to select any bandwidth as an influential attribute, but the fact that *hfd08* is the best feature for both datasets implies an important correlation of the fractal dimension with the electrode number 8.

The following two tables present GNB's prediction accuracies and feature importance analysis for partition S.

Table 6. Partition S prediction accuracies for GNB.

Patient	Window length (s)	Prediction accuracy without SKB (%)	Prediction accuracy with SKB (%)
1	60	62	71
1	10	60	63
2	60	57	70
2	10	53	63

These results show a decline in accuracy compared to the partition W results. As windows from different segments are used in the training and test set, a lower accuracy was expected.

Table 7 shows the ten features selected by SKB for partition S along with the prediction results presented in the previous table.

Table 7. The selected 10 features and prediction accuracies using SKB for partition S.

	Patient 1 (60 s)	Patient 1 (10 s)	Patient 2 (60 s)	Patient 2 (10 s)
1.	psr_alpha07 (121.4)	psr_theta06 (597.1)	hfd13 (144.1)	hfd08 (1033.1)
2.	psr_theta15 (107.0)	psr_alpha07 (590.9)	hfd08 (129.2)	hfd13 (838.0)

3.	psr_theta14 (100.1)	psr_theta15 (567.5)	hfd10 (114.2)	hfd16 (836.9)
4.	psr_theta06 (99.8)	psr_theta14 (535.9)	hfd06 (109.8)	hfd06 (780.9)
5.	psr_theta13 (88.6)	psr_theta13 (503.4)	hfd05 (97.9)	hfd05 (755.3)
6.	psr_theta07 (83.3)	psr_theta05 (444.2)	hfd16 (93.8)	hfd15 (720.6)
7.	psr_theta05 (75.5)	psr_alpha06 (434.2)	hfd24 (89.2)	hfd24 (719.4)
8.	psr_alpha12 (60.4)	psr_theta07 (398.7)	hfd07 (87.0)	hfd23 (629.2)
9.	psr_alpha06 (59.7)	psr_alpha05 (371.1)	hfd14 (84.2)	hfd10 (597.8)
10.	psr_theta12 (49.4)	psr_theta12 (350.1)	hfd15 (81.7)	hfd22 (561.5)
%	**71**	**63**	**70**	**63**

As with partition W, only *prs_theta* and *psr_alpha* appear in the ten selected features of Patient 1. In addition, fractal dimension is the single measure contributing to the predictions of Patient 2. The gradual decline of the scores indicates that a single significant electrode cannot be pointed out as the single main determinant of predictive power. Instead it might be that certain correlations of electrodes and measures appear significant.

To summarize the performance of the classifier for different partitions, the previous results are presented in Table 8. The best measure is selected as the most frequent measure among the best features.

Table 8. Patient 1 and 2 results for GNB.

Patient	**Window length (s)**	**Partition**	**Prediction accuracy without SKB (%)**	**Prediction accuracy with SKB (%)**	**Best measure (using SKB)**
1	10	W	64	77	psr_theta
1	10	S	60	63	psr_theta
1	60	W	65	80	psr_theta
1	60	S	62	71	psr_theta
2	10	W	53	78	hfd
2	10	S	53	63	hfd
2	60	W	55	78	hfd
2	60	S	57	70	hfd

Overall, GNB behaves quite similarly for datasets of a single patient. For Patient 1, the spectral power features are the most significant, with theta being the most influential bandwidth. The fractal dimension, which measures the complexity of a time series, has a significant impact on the results of the second patient as it is the only measure selected from the Patient 2 datasets. Remarkably, the 7th and 6th electrode show a correlation with the theta and alpha band for Patient 1, appearing with both in all of the 10 second and 60 second lists of the first patient.

Also notable is the fact that the 60 second dataset's results are generally better than those of the 10 second dataset, despite the fact that the latter has more data for the classifier to work with. This implies that the 10 second windows might be too detailed and noisy for effective classification and the 60 second windows describe the data better.

The results described in this chapter were obtained using the code in the file *predict_GNB.py.*

4.2.3 Random Forest Classifier

This chapter presents the classification results and feature importance analysis for the Random Forest classifier.

The following is an analysis about the results of partition W.

Table 9 shows Random Forest classifier's (RFC) prediction results for partition W. The numbers in the parentheses following the feature names represent the mean decrease impurity values calculated by the *RandomForestClassifier* model and acquired from the model's *feature_importances_* attribute. The last row represents the prediction accuracies.

Table 9. Partition W RFC top 10 features and prediction accuracies.

	Patient 1 (60 s)	**Patient 1 (10 s)**	**Patient 2 (60 s)**	**Patient 2 (10 s)**
1.	ps_theta05 (0.173)	psr_theta15 (0.326)	hfd08 (0.426)	hfd08 (0.525)
2.	ps_beta09 (0.120)	mobility01 (0.255)	hfd06 (0.102)	ps_alpha13 (0.11)
3.	psr_alpha07 (0.103)	psr_beta15 (0.081)	ps_theta01 (0.082)	ps_alpha14 (0.089)
4.	mobility01 (0.097)	hfd03 (0.039)	ps_beta05 (0.073)	hfd04 (0.073)
5.	psr_theta15 (0.090)	ps_delta07 (0.032)	ps_highgamma04 (0.065)	ps_highgamma04 (0.034)
6.	activity03 (0.079)	psr_alpha07 (0.025)	ps_lowgamma04 (0.033)	activity09 (0.012)
7.	psr_beta15 (0.057)	hfd09 (0.022)	hfd03 (0.032)	hfd14 (0.010)

8.	ps_theta15 (0.046)	hfd01 (0.018)	ps_highgamma02 (0.021)	8. ps_lowgamma05 (0.0095)
9.	ps_delta07 (0.045)	hfd08 (0.013)	psr_lowgamma20 (0.018)	9. ps_delta09 (0.0093)
10.	psr_highgamma01 (0.018)	ps_theta05 (0.013)	ps_theta02 (0.0074)	10. ps_theta09 (0.0090)
%	**95**	**97**	**94**	**97**

In Table 9 we observe that the prediction accuracies are extremely high for both patients (above 94% in all cases).

We can concur from these importances that the classification processes for both Patient 1 datasets (10 seconds and 60 seconds) were quite similar as 6 features out of 10 are the same. This is especially remarkable due to the fact that there are 270 features for the Patient 1 datasets.

We can deduct that the theta band is the most influential for Patient 1, as it occurs twice in the 10 second dataset ranking and thrice in the 60 second list and is the bandwidth for the best feature in both cases. The most effective electrode is the 15th as it appears thrice among the minute dataset's best features and twice in the 10 second list. The 7th electrode is almost as frequent. In addition, the spectral power and its ratio appear numerous times in both lists.

For Patient 2, spectral power is the most numerous measure for both cases. Another effective measure appears to be *hfd*, particularly the feature *hfd08*, which ranks first in both feature lists. However, it is harder to distinguish the best frequency band and electrode, as the best features vary a lot for both of Patient 2 datasets. The gamma band (low gamma and high gamma) is the most frequent bandwidth in these lists but does not exhibit a large dominance.

The analysis for RFC's prediction accuracies and feature importance for partition S are presented along with Table 10.

Table 10. Partition S RFC top 10 features and prediction accuracies.

	Patient 1 (60 s)	**Patient 1 (10 s)**	**Patient 2 (60 s)**	**Patient 2 (10 s)**
1.	ps_delta07 (0.156)	mobility01 (0.177)	hfd08 (0.318)	hfd08 (0.44)
2.	ps_theta05 (0.099)	ps_theta15 (0.132)	hfd13 (0.134)	hfd06 (0.13)
3.	psr_alpha07 (0.071)	ps_delta07 (0.072)	hfd24 (0.067)	ps_alpha13 (0.051)

4.	psr_theta15 (0.070)	psr_delta07 (0.067)	hfd06 (0.052)	hfd13 (0.047)
5.	ps_theta15 (0.049)	hfd07 (0.055)	ps_beta05 (0.048)	hfd07 (0.043)
6.	ps_theta02 (0.044)	activity01 (0.043)	ps_alpha14 (0.027)	hfd04 (0.022)
7.	mobility01 (0.038)	hfd08 (0.041)	ps_lowgamma24 (0.025)	hfd14 (0.021)
8.	psr_beta15 (0.032)	mobility07 (0.039)	activity24 (0.023)	hfd23 (0.016)
9.	ps_lowgamma09 (0.031)	ps_alpha15 (0.027)	activity08 (0.022)	hfd19 (0.014)
10.	ps_alpha05 (0.018)	activity06 (0.023)	hfd03 (0.019)	skewness09 (0.012)
%	**93**	**89**	**86**	**81**

From the results in Table 10 we can observe that the prediction accuracies have noticeably decreased for many cases but are still well above chance.

The feature ranking of the second patient's 10 second dataset is more uniform than the others. The measure *hfd* clearly dominates the list, with only two other measures appearing in the ranking. The fractal dimension is the most significant measure for the 60 second dataset as well, with the spectral power features and *activity* also showing noteworthy influence.

Compared to the partition W results of Patient 1, there are notably less common significant features for the 60 second and 10 second datasets. The spectral power (ratio) features still exhibit a remarkable dominance among the features, while the Hjorth parameters activity and mobility appear more influential for the 10 second dataset. The delta band also has a larger impact for the 10 second dataset compared to the partition W, while the theta band's influence on the 60 second dataset results remains similar. However, the 7th electrode emerges as an important influence beside the 15th.

Table 11 presents the summary of the RFC results. The best measure is selected as the most frequent measure among the best features.

Table 11. Patient 1 and 2 results for RFC.

Patient	Window length (s)	Partition	Prediction accuracy (%)	Best measure
1	10	W	97	hfd
1	10	S	89	mobility
1	60	W	95	ps_theta

1	60	S	93	ps_theta
2	10	W	97	hfd
2	10	S	81	hfd
2	60	W	94	hfd
2	60	S	86	hfd

We can observe from Table 11 that the prediction results of Patient 1 were more accurate than those of Patient 2, but not by a large margin. As half of the test set samples were preictal windows, these results can be considered very accurate.

The theta band had the most effect among all of the datasets, especially for Patient 1. While the alpha and the gamma bands exhibited some significance (for Patient 2 in particular), they were not as numerous. The feature *mobility* appeared at least once in all of the best feature lists of Patient 1, and ranked as the most frequent measure once for the 10 second dataset. The electrodes 7 and 15 are the most prominent for Patient 1, while Patient 2 does not show any affinity for a single electrode, with the 4th appearing only three times for partition W.

The fractal dimension also appears to have a great significance, especially for Patient 2, as the measure appears in nearly all of the best feature lists and *hfd08* takes the first rank four times.

While *hfd* is the measure, which dominates the Patient 2 lists, the spectral power (ratio) features are more important for Patient 1. Mostly in combination with the theta, delta and alpha bands, these features are very frequent in the Patient 1 dataset rankings.

It must be noted that these far more accurate prediction results compared to the winning solution of the Kaggle competition obtained by Tieng et al. were probably achieved due to the more balanced training and test sets used in this Thesis and a larger temporal proximity between the training and test set windows.

The code, which was used to obtain the results described in this chapter, is in the *predict_RFC.py* file.

4.2.4 Logistic Regression Classifier

This chapter presents the prediction accuracies and feature importance analysis for the Logistic Regression classifier.

Table 12 shows the Logistic Regression classifier's (LRC) prediction results for partition W. The numbers in the parentheses represent the mean coefficients that were obtained by summing up the absolute values of the coefficients of every fold and then dividing the sum with the number of folds.

Table 12. Partition W LRC top 10 features and prediction accuracies.

	Patient 1 (60 s)	**Patient 1 (10 s)**	**Patient 2 (60 s)**	**Patient 2 (10 s)**
1.	psr_theta07 (19.6)	psr_theta07 (65.7)	psr_alpha14 (15.9)	psr_alpha14 (99.3)
2.	psr_alpha07 (17.0)	hfd01 (63.5)	psr_alpha06 (12.6)	psr_alpha06 (93.8)
3.	psr_theta15 (8.9)	hfd15 (53.0)	psr_highgamma01 (10.9)	psr_alpha24 (84.8)
4.	psr_theta06 (8.9)	psr_theta11 (46.6)	psr_lowgamma11 (10.6)	psr_alpha13 (80.1)
5.	psr_alpha02 (8.2)	psr_alpha07 (42.7)	psr_alpha13 (10.6)	psr_alpha01 (73.7)
6.	psr_theta11 (6.5)	hfd08 (38.3)	psr_alpha22 (9.9)	psr_alpha17 (62.6)
7.	psr_theta14 (6.2)	hfd07 (30.2)	psr_alpha24 (8.9)	psr_highgamma19 (62.4)
8.	psr_theta13 (6.1)	hfd05 (29.8)	psr_alpha05 (8.3)	psr_lowgamma19 (62.1)
9.	psr_theta02 (6.1)	psr_beta15 (28.5)	psr_alpha21 (8.2)	psr_beta02 (54.8)
10.	psr_highgamma08 (5.9)	hfd10 (27.8)	psr_beta11 (7.7)	psr_alpha05 (53.9)
%	**96**	**97**	**96**	**96**

Table 12 shows extremely high prediction accuracies for LRC, with every result being at least 96%.

We can see in Table 12 that the spectral power (ratio) features are the most common among the dataset lists, with the theta and alpha bandwidths being the most prominent. The measure *psr_alpha* is very dominant in the second patient's lists, while *psr_theta* is a very common measure in the Patient 1 rankings. Although the 7th electrode seems to be the most significant electrode for Patient 1, a generally influential electrode cannot be stated based on these results.

Table 13 presents LRC's prediction results and feature importance analysis for partition S.

Table 13. Partition S LRC top 10 features and prediction accuracies.

	Patient 1 (60 s)	**Patient 1 (10 s)**	**Patient 2 (60 s)**	**Patient 2 (10 s)**
1.	psr_theta06 (9.91)	mobility03 (34.0)	psr_alpha09 (7.25)	hfd13 (13.10)
2.	psr_theta07 (8.72)	mobility11 (32.2)	psr_alpha14 (6.99)	hfd24 (8.06)
3.	psr_theta15 (8.09)	mobility08 (30.9)	psr_alpha06 (6.70)	hfd17 (7.81)
4.	hfd15 (4.82)	mobility15 (28.9)	psr_alpha13 (6.66)	hfd21 (7.53)
5.	psr_alpha07 (4.17)	mobility13 (28.4)	psr_highgamma02 (6.35)	hfd20 (7.27)
6.	psr_theta05 (4.13)	mobility04 (28.4)	psr_alpha05 (5.54)	psr_alpha06 (5.89)
7.	psr_alpha02 (4.07)	mobility01 (27.6)	psr_theta24 (5.26)	hfd05 (5.21)
8.	psr_highgamma08 (3.77)	mobility07 (27.2)	psr_lowgamma01 (5.04)	hfd11 (5.09)
9.	psr_alpha03 (3.43)	mobility09 (27.0)	psr_alpha01 (4.94)	hfd03 (4.57)
10.	psr_theta13 (2.94)	mobility10 (26.9)	psr_highgamma09 (4.58)	hfd07 (4.44)
%	**87**	**92**	**78**	**79**

In Table 13 we can observe that the results of the Patient 1 datasets are significantly more accurate than the results of Patient 2, which have declined considerably compared to partition W.

The feature importance results also show a noticeable change from those of partition W. While previously the significant features of the first patient's 10 second dataset included various measures like *hfd* and spectral power ratio of several different bands, *mobility* is the only measure among the ten best in this case and gives the best prediction accuracy, 92%, among the four datasets. However, as mobility is a measure that can be interpreted in the frequency domain as an estimate of the mean frequency, this is not entirely surprising.

The 60 second datasets show an affinity for the spectral power ratio measure, with the only other feature for Patient 1 being *hfd15*. Alpha is the most prominent bandwidth for Patient 2, while the theta band has a larger influence on the Patient 1 results, similarly to partition W. The fractal dimension again exhibits a significant influence for the Patient 2 10 second dataset.

As with partition W, the 7th and 15th electrode appear numerous times in the Patient 1 best feature lists.

Table 14 summarizes the LRC results.

Table 14. Patient 1 and 2 results for LRC.

Patient	Window length (s)	Partition	Prediction accuracy (%)	Best measure
1	10	W	97	hfd
1	10	S	92	mobility
1	60	W	96	psr_theta
1	60	S	87	psr_theta
2	10	W	96	psr_alpha
2	10	S	79	hfd
2	60	W	96	psr_alpha
2	60	S	78	psr_alpha

In Table 14 we can observe that when comparing the results of partition W and S with the same datasets, the accuracy declines far less with Patient 1 than with Patient 2.

The most significant measures for Patient 1 vary greatly, as opposed to Patient 2, where almost all the best measures are *psr_alpha*.

Although the 7th and 15th electrode showed an impact on the Patient 1 results, an overall significant electrode does not appear from these results.

The results described in this chapter were acquired using the code in the file *predict_LRC.py*.

4.3 Summary

This chapter will summarize the results presented in the previous subchapters, first describing results for both patients separately and then presenting a comparison of the average scores over different attributes.

4.3.1 Patient 1

Table 15 is a summary of all the predictions for Patient 1 and the analyzation in the previous chapters. The best bandwidth, measure and electrodes are selected as the modes from all of the values in the corresponding results.

Table 15. Patient 1 results summary.

Partition	Window length (s)	Algorithm	Accuracy (%)	Best measure	Best bandwidth	Best electrodes
W	10	GNB	64	-	-	-
W	10	GNB + SKB	77	psr_theta	theta	6, 7, 12
W	10	RFC	97	hfd	theta	7, 15
W	10	LRC	97	hfd	theta	7
W	60	GNB	65	-	-	-
W	60	GNB + SKB	80	psr_theta	theta	6, 7, 12
W	60	RFC	95	ps_theta	theta	15
W	60	LRC	96	psr_theta	theta	7
S	10	GNB	60	-	-	-
S	10	GNB + SKB	63	psr_theta	theta	5, 6, 7
S	10	RFC	89	mobility	delta	7
S	10	LRC	92	mobility	-	-
S	60	GNB	62	-	-	-
S	60	GNB + SKB	71	psr_theta	theta	6, 7, 12
S	60	RFC	93	ps_theta	theta	7, 15
S	60	LRC	87	psr_theta	theta	7, 15

From Table 15, we can observe that both RFC and LRC successfully produced results that are above the statistical base method GNB's results. The average accuracy over all of the experiments is 80.5%.

These results show that the most important band is undoubtedly theta, while the alpha and delta bands also provide some influence on the results.

The most significant electrode is the 7th, with electrode number 6 also making a notable impact.

Overall, the most important measures were the spectral powers within the most significant bandwidths, the fractal dimension and the Hjorth mobility.

4.3.2 Patient 2

Table 16 is a summary of all the predictions for Patient 2 and the analyzation in the previous chapters. The best bandwidth, measure and electrodes are selected as the modes from all the values in the corresponding results.

Table 16. Patient 2 results summary.

Partition	Window length (s)	Algorithm	Accuracy (%)	Best measure	Best bandwidths	Best electrodes
W	10	GNB	53	-	-	-
W	10	GNB + SKB	78	hfd	-	-
W	10	RFC	97	hfd	alpha, gammas	9
W	10	LRC	96	psr_alpha	alpha	19
W	60	GNB	55	-	-	-
W	60	GNB + SKB	78	hfd	-	-
W	60	RFC	94	hfd	theta, gammas	2, 4
W	60	LRC	96	psr_alpha	alpha	-
S	10	GNB	53	-	-	-
S	10	GNB + SKB	63	hfd	-	-
S	10	RFC	81	hfd	-	13
S	10	LRC	79	hfd	-	-
S	60	GNB	57	-	-	-
S	60	GNB + SKB	70	hfd	-	-
S	60	RFC	86	hfd	-	8
S	60	LRC	78	psr_alpha	alpha	1, 9

In Table 16 we can observe that the prediction results of Patient 2 are less accurate than those of Patient 1, but still have a fairly high 75.9% average score.

For Patient 2, there were a lot of cases where the best feature list was dominated by measures, which did not allow to make statements about the best bandwidth or electrode and thus, many empty slots are in this table.

The most important measure is the Higuchi fractal dimension, which appeared the most frequently in almost all of the best feature lists, with the only other stand-out measure being *psr_alpha*.

The electrodes do not appear to show any patterns, varying for almost all of the combinations.

4.3.3 Average Scores

This chapter presents a comparative analysis of all the prediction accuracies over three different attributes: patients, window lengths, and partitions.

Figure 9 presents a comparison of average scores for the patients.

The statistical GNB has considerably lower scores than the more complex methods, with RFC outperforming every method with a total average score of 91.5%, a little over a percent larger than the average score of LRC. The scores of Patient 1 are higher than those of Patient 2 in all cases, the gap between the scores being the smallest for the GNB and SKB combination and biggest for GNB alone.

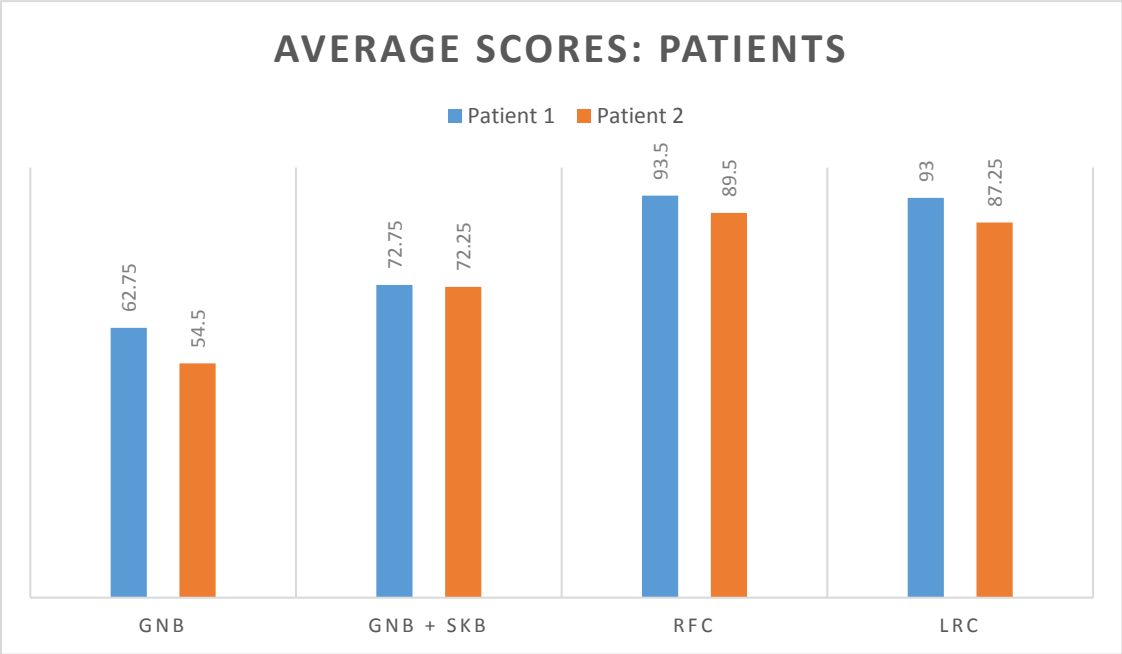

Figure 9. The average scores for patients 1 and 2.

Figure 10 depicts the average scores for datasets with different window lengths. In the figure we can observe that in most cases the minute length windows perform better than the shorter time windows. Only with LRC does this tendency change to the opposite.

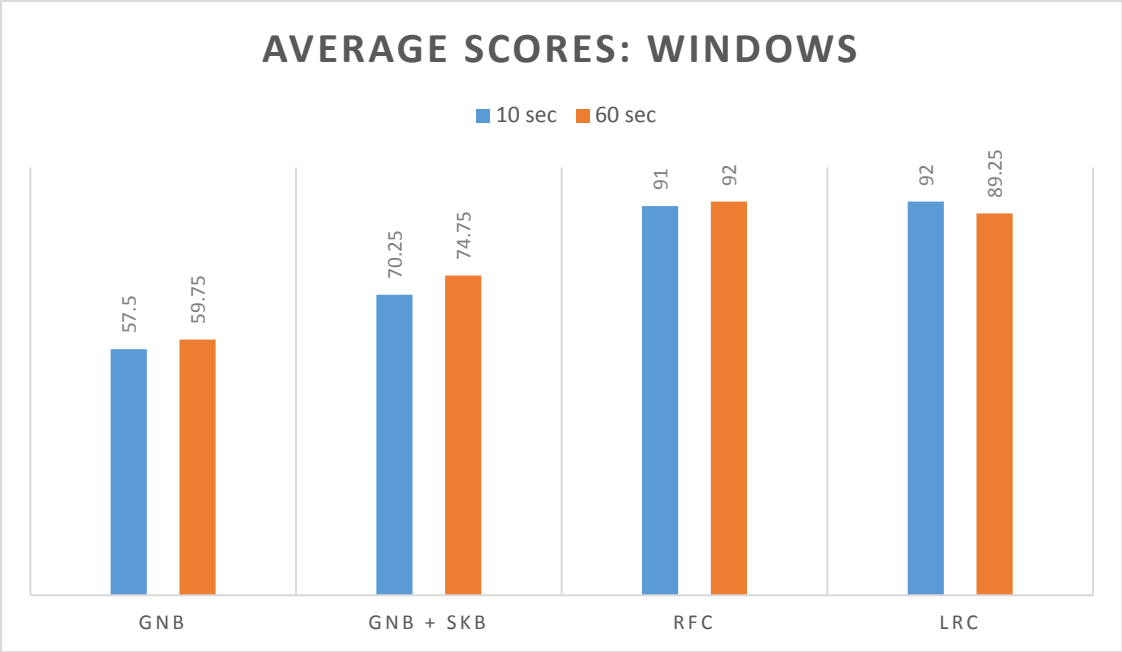

Figure 10. The average scores for 10 second and 60 second datasets.

Figure 11 shows the average scores of the two partitions. This figure indicates that overall, explicitly including windows from different segments did affect the prediction scores negatively, as was expected. The largest decline is with LRC, while the results of GNB hardly changed.

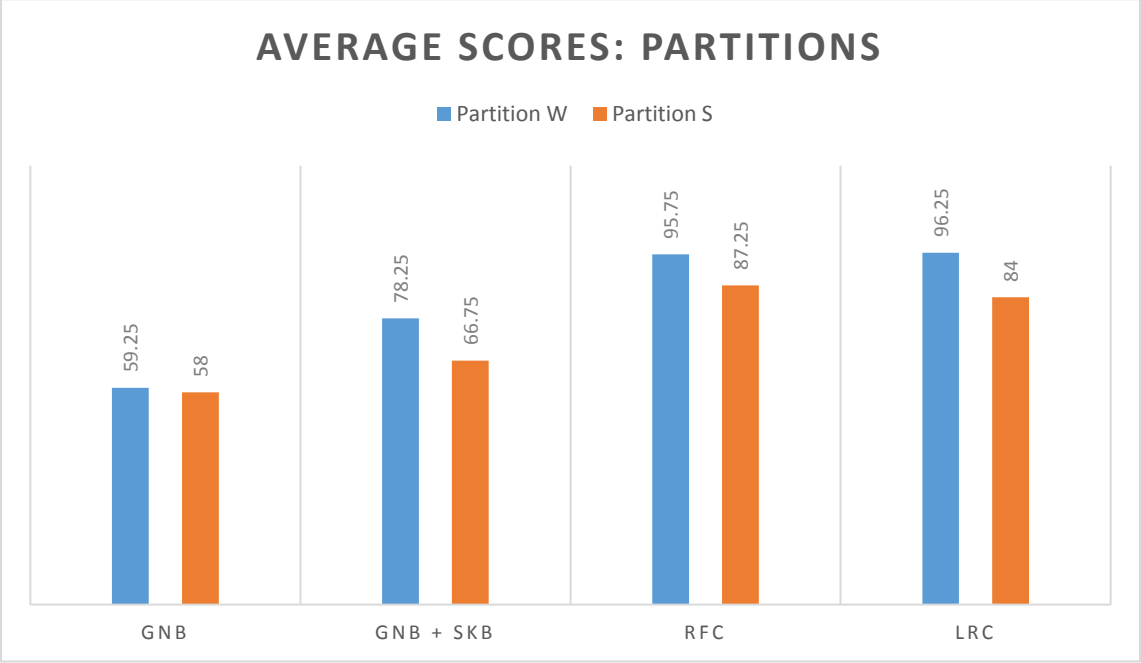

Figure 11. The average scores for partitions 1 and 2.

From these results we can concur that in general, RFC and LRC were equally successful, producing scores that greatly exceeded those of GNB, with or without SKB. Explicitly including windows from random different segments to the training and test data had the most impact on the results, compared to the effect of different patients and window lengths.

5. Discussion

This chapter describes the limitations of the results obtained in this Thesis and suggests different ideas for future research that these results might be helpful for.

5.1 Limitations

The results obtained in this Thesis are limited to only two human subjects and a little over 20 hours of iEEG data. As it appeared in the results, different patients have different measures, features that are most significant for these classifiers and to find any general patterns, a considerably larger amount of iEEG data for a larger variety of patients is necessary.

Also, as intracranial EEG does not have a standard for electrode placement, the locations of the electrodes that recorded these datasets is unknown and thus, no conclusions can be made in relation to the seizure origins' locations.

In addition, only a single feature importance evaluation method was tested with each classifier and some other measures of sensitivity like knock-out analysis (excluding one feature to measure its impact on the results) might have provided additional insight. This was decided against due to limited time.

5.2 Future Work

This Thesis presented the most important measures found by conducting experiments but further biological analysis of these results would be an important future development. By drawing parallels with the biological aspects of the human brain and epileptic seizure generation, these results could give further insight into the brain activity connected to seizures.

6. Summary

The aim of this Thesis was to **analyze the importance** of features used in epileptic seizure prediction. These features were extracted from the data by calculating the values of **18 measures** that are commonly used in seizure prediction. These measures were used to extract features from **iEEG data** of two patients. A **moving window analysis** was conducted on every electrode's data record and the 18 extracted features of every concurrent window were concatenated to form a single data row in the new dataset. Two datasets were calculated for both patients: one with 10 second windows and the other with minute-length windows.

The 18 measures are the following:

- Hjorth activity, mobility and complexity,
- Higuchi fractal dimension,
- skewness,
- kurtosis,
- spectral power for delta, theta, alpha, beta, low gamma and high gamma frequency bands, and
- spectral power in each of the previously mentioned frequency bands normalized by total power in all of the frequency bands.

In addition, the impact of the different frequency bands and the electrodes that recorded the data were analyzed.

Three classifiers of the scikit-learn library were used for machine learning and predicting: **Random Forest Classifier** (RFC), **Logistic Regression Classifier** (LRC), and **Gaussian Naïve Bayes Classifier** (GNB). Other classification methods were tried as well (Support Vector Machines, Stochastic Gradient Descent, and Multi-layer Perceptron), which provided a smaller classification performance and therefore were not suitable for feature importance analyses.

To analyze the feature importances, a different technique was used for every classifier. For RFC, feature importances were acquired using mean decrease impurity, and for LRC, the absolute values of the feature coefficients were calculated. The scikit-learn library's SelectKBest method was used in combination with GNB to get the most important features.

All of the methods were used with **cross-validation** of 5 folds. The training and test set contained an equal number of randomly chosen preictal and interictal windows.

RFC and LRC in particular produced extremely accurate results, while the statistical GNB had moderate to good results. The prediction results were more accurate with Patient 1 than with Patient 2 and the temporally less correlated partition S had a smaller average score than the more correlated partition W. Overall, the 60 second datasets produced more accurate results with the exception of LRC.

For Patient 1, the **spectral power (ratio) features** proved to be the most impactful and the **theta band** was remarkably dominant among the bandwidths, with also the **delta** and **alpha** bands making a notable contribution.

The **Higuchi fractal dimension** was the most important feature for Patient 2, with only the spectral power ratio of the alpha band appearing as another significant feature. A significant feature, which ranked among the top two measures for both patients several times, was *hfd08* – the fractal dimension from the recording of the 8th electrode.

Analysis of predictive features is important in order to understand the mechanisms behind epileptic seizures and to improve the selection of features for seizure prediction. The ultimate goal of seizure prediction is to create a reliable system that could predict seizures on real time data and warn patients of an oncoming seizure. As predicting can be a slow process, especially with vast amounts of EEG data, a smart selection of features has to be made to speed up the classification process and increase the accuracy of the algorithms. This cannot be done, if the influences of features are not measured in studies.

Future work with this Thesis would be an in-depth biological analysis of these results, finding correlations between the significant measures and epileptic seizure generation.

Bibliography

[1] P. O. Shafer and J. I. Sirven, "Epilepsy Statistics," 2013. [Online]. Available: http://www.epilepsy.com/learn/epilepsy-statistics. [Accessed 24 March 2017].

[2] "Epilepsy," World Health Organization, [Online]. Available: http://www.who.int/mediacentre/factsheets/fs999/en/. [Accessed 24 March 2017].

[3] "Seizure forecasting systems hold promise for improving the quality of life for patients with epilepsy," Kaggle, [Online]. Available: https://www.kaggle.com/c/seizure-prediction. [Accessed 24 March 2017].

[4] S. A. Kumar, L. Nigam, D. Karnam, S. K. Murthy, P. Fedorovych and V. Kalidindi, "Machine Learning for Seizure Prediction: A Revamped Approach," in *2015 International Conference on Advances in Computing, Communications and Informatics* , Kochi, 2015.

[5] "Brain," National Geographic, [Online]. Available: http://www.nationalgeographic.com/science/health-and-human-body/human-body/brain/. [Accessed 25 April 2017].

[6] N. Sivasankari, K. Thanushkodi and H. K. Naidu, "An Extensive Review of Significant Researches on Epileptic Seizure," in *Advances in Biomedical Research*, World Scientific and Engineering Academy and Society, 2010, pp. 330-353.

[7] S. C. Schachter, P. O. Shafer and J. I. Sirven, "What Causes Epilepsy and Seizures?," Epilepsy Foundation, July 2013. [Online]. Available: http://www.epilepsy.com/learn/epilepsy-101/what-causes-epilepsy-and-seizures. [Accessed 25 April 2017].

[8] M. Demitri, "Types of Brain Imaging Techniques," Psych Central, 17 May 2016. [Online]. Available: https://psychcentral.com/lib/types-of-brain-imaging-techniques/. [Accessed 27 April 2017].

[9] "WHAT ARE BRAINWAVES?," Brainworks, [Online]. Available: http://www.brainworksneurotherapy.com/what-are-brainwaves. [Accessed 29 April 2017].

[10] L. Kuhlmann, "Getting Started in the Seizure Prediction Competition: Impact, History, & Useful Resources," Kaggle, 14 October 2016. [Online]. Available: http://blog.kaggle.com/2016/10/14/getting-started-in-the-seizure-prediction-competition-impact-history-useful-resources/. [Accessed 29 April 2017].

[11] J. P. Lachaux, D. Rudrauf and P. Kahane, "Intracranial EEG and human brain mapping," *Journal of Physiology,* vol. 97, no. 4-6, pp. 613-628, 2003.

[12] A. Yadollahpour and M. Jalilifar, "Seizure Prediction Methods: A Review of the Current Predicting Techniques," *Biomedical & Pharmacology Journal,* vol. 7(1), pp. 153-162, 2014.

[13] C.-Y. Chiang, N.-F. Chang, T.-C. Chen, H.-H. Chen and L.-G. Chen, "Seizure Prediction Based on Classification of EEG Synchronization Patterns with On-line Retraining and Post-Processing Scheme," *33rd Annual International Conference of the IEEE EMBS,* pp. 7564-7569, 3 September 2011.

[14] "Kaggle: American Epilepsy Society Seizure Prediction Challenge," Kaggle, 2014. [Online]. Available: https://www.kaggle.com/c/seizure-prediction/data. [Accessed February 2017].

[15] Q. M. Tieng, M. Chen, S. C. Bosshard, D. W. Abbot and P. C. Adkins, "QMSDP on Seizure Prediction," 2014.

[16] R. Vicente Zafra, Writer, *Lecture 4: Data Analysis I.* [Performance]. Introduction to Computational Neuroscience, 2016.

[17] B. Hjorth, "EEG analysis based on time domain properties," *Electroencephalography and Clinical Neurophysiology,* vol. 29, no. 3, pp. 306-310, 1970.

[18] G. Giannakakis, V. Sakkalis, M. Pediaditis and M. Tsiknakis, "Methods for Seizure Detection and Prediction: An Overview," in *Modern Electroencephalographic Assessment Techniques,* Springer New York, 2015, pp. 131-157.

[19] F. Mormann, T. Kreuz, C. Rieke and K. Lehnertz, "On the predictability of epileptic seizures," *Clinical Neurophysiology,* pp. 569-587, 2005.

[20] M. Kaboli, A. De La Rosa T, R. Walker and G. Cheng, "In-Hand Object Recognition via Texture Properties with Robotic Hands, Artificial Skin, and Novel Tactile Descriptors," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids),* Seoul, 2015.

[21] F. Bao, X. Liu and C. Zhang, "PyEEG: An Open Source Python Module for EEG/MEG Feature Extraction," *Computational Intelligence and Neuroscience,* 2011.

[22] R. Esteller, G. Vachtsevanos, J. Echauz and B. Litt, "A comparison of waveform fractal dimension algorithms," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 2, pp. 177-183, 2001.

[23] S. Dean and B. Illowsky, "Descriptive Statistics: Skewness and the Mean, Median, and Mode," OpenStax, [Online]. Available: http://cnx.org/contents/bE-w34Vi@9/Descriptive-Statistics-Skewnes. [Accessed 1 May 2017].

[24] "Note on Skewness with Example," [Online]. Available: https://www.kullabs.com/classes/subjects/units/lessons/notes/note-detail/9958. [Accessed 2 May 2017].

[25] Massachusetts Institute of Technology, "Chapter 3 A statistical description of turbulence," [Online]. Available: https://ocw.mit.edu/courses/earth-atmospheric-and-planetary-sciences/12-820-turbulence-in-the-ocean-and-atmosphere-spring-2007/lecture-notes/ch3.pdf. [Accessed 1 May 2017].

[26] J. Rasekhi, M. R. K. Mollaei, M. Bandarabadi, C. A. Teixeira and A. Dourado, "Preprocessing effects of 22 linear univariate features on the performance of seizure prediction methods," *Journal of Neuroscience Methods,* 2013.

[27] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579-2605, 2008.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel , B. Thirion and O. Grisel, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* 2011.

[29] T. M. Mitchell, Writer, *Gaussian Naïve Bayes, and Logistic Regression.* [Performance]. Machine Learning Department, Carnegie Mellon University, 2010.

[30] "1.9. Naive Bayes," [Online]. Available: http://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes. [Accessed 30 April 2017].

[31] "sklearn.naive_bayes.GaussianNB," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB. [Accessed 2 May 2017].

[32] C. Roach, "Building Decision Trees in Python," O'Reilly Media, Inc., 9 February 2006. [Online]. Available: http://www.onlamp.com/pub/a/python/2006/02/09/ai_decision_trees.html. [Accessed 29 April 2017].

[33] M. Galea, "Decision Trees with Kotlin," 20 August 2016. [Online]. Available: http://cloudmark.github.io/Decision-Trees/. [Accessed 29 April 2017].

[34] T. K. Ho, "Random Decision Forests," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, 1995.

[35] "sklearn.ensemble.RandomForestClassifier," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed 30 April 2017].

[36] D. Jurafsky and J. H. Martin, Speech and Language Processing, 2016.

[37] J. Brownlee, "Logistic Regression for Machine Learning," 1 April 2016. [Online]. Available: http://machinelearningmastery.com/logistic-regression-for-machine-learning/. [Accessed 30 April 2017].

[38] "sklearn.linear_model.LogisticRegressionCV," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html. [Accessed 30 April 2017].

[39] "Python for Data Analysis Part 26: Analysis of Variance (ANOVA)," 23 November 2015. [Online]. Available: http://hamelg.blogspot.com.ee/2015/11/python-for-data-analysis-part-16_23.html. [Accessed 5 May 2017].

[40] A. Saabas, "Selecting good features – Part III: random forests," 1 December 2014. [Online]. Available: http://blog.datadive.net/selecting-good-features-part-iii-random-forests/. [Accessed 2 May 2017].

Appendices

I. License

Non-exclusive license to reproduce thesis and make thesis public

I, Mari Liis Velner,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

Analyzing Predictive Features of Epileptic Seizures in Human Intracranial EEG Recordings,

supervised by Raul Vicente Zafra, PhD,

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **11.05.2017**