Automatic Sleep Stage Scoring through single channel EEG data

by

Martin Sofroniev

GRADUATION REPORT

Submitted to

Hanze University of Applied Sciences Groningen

in partial fulfillment of the requirements

for the degree of

Fulltime Master Sensor System Engineering

2018

ABSTRACT

Automatic Sleep Stage Scoring through single channel EEG data

by

Martin Sofroniev

(to be written…)

DECLARATION

I hereby certify that this report constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the report describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Martin Sofroniev

ACKNOWLEDGEMENTS

(to be written…)

(leaving this page blank because of standard…)

TABLE OF CONTENTS

[CHAPTER I 5](#_Toc515823151)

[RATIONALE 5](#_Toc515823152)

[CHAPTER II 8](#_Toc515823153)

[SITUATIONAL & THEORETICAL ANALISYS 8](#_Toc515823154)

[2.1 Sleep and sleep studies 9](#_Toc515823155)

[2.1.1 Effects of sleep on the human body and behavior 9](#_Toc515823156)

[2.1.2 EEG signal and the R&K and AASM systems of classification 9](#_Toc515823157)

[2.1.3 Circadian rhythm and Sleep homeostat 11](#_Toc515823158)

[2.2 Machine Learning 12](#_Toc515823159)

[2.2.1 Overview 12](#_Toc515823160)

[2.2.2 Methods for feature extraction 13](#_Toc515823161)

[2.2.2.1 Time-domain 13](#_Toc515823162)

[2.2.2.2 Frequency domain 15](#_Toc515823163)

[2.2.2.3 Time-frequency domain 20](#_Toc515823164)

[2.2.2.4 Non-linear 25](#_Toc515823165)

[2.2.3 Classifier types 27](#_Toc515823166)

[CHAPTER III 34](#_Toc515823167)

[CONCEPTUAL MODEL 34](#_Toc515823168)

[CHAPTER IV 37](#_Toc515823169)

[RESEARCH DESIGN 37](#_Toc515823170)

[4.1 Criteria, data, and equipment 38](#_Toc515823171)

[4.1.1 Criteria and equipment 38](#_Toc515823172)

[4.1.2 Data 38](#_Toc515823173)

[Hut lab Dataset 38](#_Toc515823174)

[DREAMS dataset 38](#_Toc515823175)

[4.2 Experiments 39](#_Toc515823176)

[4.2.1 Feature sets evaluation over different channels 39](#_Toc515823177)

[4.2.2 Evaluation of different ML algorithms 39](#_Toc515823178)

[4.2.3 Individual feature evaluation 39](#_Toc515823179)

[4.2.4 Hidden Markov Model evaluation 39](#_Toc515823180)

[4.3 Hypotheses 40](#_Toc515823181)

[4.3.1 Time domain feature hypothesis: 40](#_Toc515823182)

[4.3.2 Frequency features 41](#_Toc515823183)

[4.3.3 Time-frequency features 41](#_Toc515823184)

[4.3.4 ML algorithm performance 41](#_Toc515823185)

[CHAPTER V 42](#_Toc515823186)

[RESEARCH RESULTS 42](#_Toc515823187)

[5.1 Feature set evaluation and channel evaluation 43](#_Toc515823188)

[5.1.1 10 Second Hut Lab data 43](#_Toc515823189)

[5.1.2 Dreams dataset data 47](#_Toc515823193)

[5.2 Machine learning algorithm testing 47](#_Toc515823194)

[5.2.1 Random Forest 47](#_Toc515823195)

[5.2.2 K-nearest Neighbor 47](#_Toc515823196)

[5.2.3 Support vector Machine 47](#_Toc515823197)

[5.3 Feature evaluation 47](#_Toc515823198)

[5.3.1 10 seconds Hut data 47](#_Toc515823199)

[5.3.2 Dreams dataset 47](#_Toc515823200)

[5.4 Final algorithm check with PCA, HMM, one model on separate database comparison 47](#_Toc515823201)

[CHAPTER VI 48](#_Toc515823202)

[CONCLUSIONS & RECMMENDATIONS 48](#_Toc515823203)

[LIST OF DEFINITIONS AND ABBREVIATIONS 50](#_Toc515823204)

[REFERENCES 51](#_Toc515823205)

[APPENDIX A 54](#_Toc515823206)

LIST OF TABLES

LIST OF FIGURES

[Figure 1 The 10-20 system for EEG electrode placement 9](file:///G:\Master%20Graduation\Master%20thesis\report%20parts\%5b27-05-2018%5d%20%5bMaster%20Graduation%20Report%5d%20Automatic%20Sleep%20Stage%20Scoring%20through%20single%20channel%20EEG%20data.docx#_Toc515823207)

[Table 1 Comparison of sleep stage classification standards [4], [5] 10](#_Toc515823208)

[Figure 2. Frequency bands of the sleep EEG signal represented as signals in time.[26] 10](#_Toc515823209)

[Figure 3. K-complexes and Sleep spindles as parts of an EEG signal in time 11](#_Toc515823210)

[Figure 4. A hypnogram showing sleep stages and cycles in time [30] 12](#_Toc515823211)

[Figure 5. Data visualisation under different feature spaces. Moving from a lower dimension feature space (left) to a higher dimension feature space (right) can make it easier to segregate the classes. 12](#_Toc515823212)

[Figure 6. First 4 statistical moments visualization and explanation 14](#_Toc515823213)

[Figure 7. The zero-crossing rate of a time-invariant signal. Notice how the number of zero-crossings is not representative of the frequency of the signal 14](#_Toc515823214)

[Figure 8 A time varying signal and its DFT magnitude plot. 16](#_Toc515823215)

[Figure 9 Half a sine wave and its DFT magnitude plot. 17](#_Toc515823216)

[Figure 10. Frequency power spectrum of an aperiodic signal with colour coded spans of the frequency bands used for sleep classification. 18](#_Toc515823217)

[Figure 11. A generated AR model PSD output (blue) and its estimation based on the coefficients (n=4) extracted by using the Yule-Walker equations 19](#_Toc515823218)

[Figure 12. Evolution of the AR parameters with increasing **i** term 20](#_Toc515823219)

[Figure 13. Visualization of the uncertainty principle. On the left the oscillating wave presents more information about the frequency (momentum) of wave (particle) and insufficient information about its position, and on the right the pulse wave presents much more information about the position of the particle but less about the frequency. 21](#_Toc515823220)

[Figure 14. A representation of a rectangular window sliding over a time-series. In STFT this window defines the limits of the Fourier transform as the signal outside of the window is ignored. 21](#_Toc515823221)

[Figure 15. Different window functions represented in the area they cover in time-series representation (blue) and the Fourier transform of a waveform under the respective window function (orange) 22](#_Toc515823222)

[Figure 16. A Nuttall window (red) over a time-series (blue) without overlapping (left) shows how much of the signal remains in the zero-valued are (outside of the window). On the right it is shown how overlapping the windows resolves this issue as almost all of the signal is under the area covered by the windows (50% overlap) with the amplitude flatness shwon in green.. 23](#_Toc515823223)

[Figure 17. Spectrogram of the STFT of a time-varying signal with 4 distinct segments (5Hz, 10hz, 15Hz, 20Hz) of equal length (n=512) with varying window length (red N annotation). We can see how increasing the number of samples of a window decreases the temporal information (seen by the overlap of the segments and the decrease of colour contrast). 23](#_Toc515823224)

[Figure 18. A visual representation of how the CWT algorithm works where **t** indicates time step and **s** indicates scale. First a low scale is selected (compressed mother wavelet) and it is being time shifted as to be compared to every temporal bin of the signal. Afterwards a new scale is selected and the process repeated. 24](#_Toc515823225)

[Figure 19. An example bin representation of the spectrograms of STFT (left) and CWT3 (right). It is apparent how the STFT has a linear distribution while the CWT allows for higher resolution at specific areas 25](#_Toc515823226)

[Figure 20. Conventional view of dimensionality and its scaling characteristic where D is the number of dimensions, I the amount of initial separations, and N the amount of resulting separations. 26](#_Toc515823227)

[Figure 21. The first four iterations of the Koch snowflake, a typical example of a fractal. 27](#_Toc515823228)

[Figure 22. Support Vector Machine classification by creating a separating hyperplane in the feature space. 28](#_Toc515823229)

[Figure 23. KNN algorithm under varying values k fro number of neighbours and the consequences of these changes 28](#_Toc515823230)

[Figure 24. Decision tree schematic where the circles represent a single binary condition. 29](#_Toc515823231)

[Figure 25. A Markov Model o a simplified weather process with the states and their switch probabilities given on the left and summarised in a state transition matrix A on the right 30](#_Toc515823232)

[Figure 26. Markov Model with probabilities of switching states given in matrix A and the probabilities of emitting an observation from a given state given in matrix B 30](#_Toc515823233)

[Figure 27 Characteristics of the RF model performance under the time domain features. 43](file:///G:\Master%20Graduation\Master%20thesis\report%20parts\%5b27-05-2018%5d%20%5bMaster%20Graduation%20Report%5d%20Automatic%20Sleep%20Stage%20Scoring%20through%20single%20channel%20EEG%20data.docx#_Toc515823234)

[Figure 28 Characteristics of the RF model performance under the frequency domain band features 44](file:///G:\Master%20Graduation\Master%20thesis\report%20parts\%5b27-05-2018%5d%20%5bMaster%20Graduation%20Report%5d%20Automatic%20Sleep%20Stage%20Scoring%20through%20single%20channel%20EEG%20data.docx#_Toc515823235)

[Figure 29 Characteristics of the RF model performance under the frequency domain AR features. 45](file:///G:\Master%20Graduation\Master%20thesis\report%20parts\%5b27-05-2018%5d%20%5bMaster%20Graduation%20Report%5d%20Automatic%20Sleep%20Stage%20Scoring%20through%20single%20channel%20EEG%20data.docx#_Toc515823236)

[Figure 30. Characteristics of the RF model performance under the continuous wavelet features 46](file:///G:\Master%20Graduation\Master%20thesis\report%20parts\%5b27-05-2018%5d%20%5bMaster%20Graduation%20Report%5d%20Automatic%20Sleep%20Stage%20Scoring%20through%20single%20channel%20EEG%20data.docx#_Toc515823237)

[Figure 31 Feature importance based on RF. 47](#_Toc515823238)

CHAPTER I

RATIONALE

The American Academy of Sleep Medicine (AASM) recommends a 7 to 9-hour duration of daily sleep in human adults [1], which would mean that humans are supposed to (on average) spend a third of their lives sleeping. Even though this might seem disproportionate, there is strong empirical evidence suggesting that spending fewer than 7 hours sleeping leads to negative physiological and behavioral effects. It has been reported that sleep deprivation has negative consequences on the endocrine functions, the metabolic and inflammatory responses, and the cardiovascular system in general. Additionally, sleep restriction negatively affects cognitive performance [2]. However, the function of sleep is debated and the process not fully understood.

In order to change that, researchers have been performing sleep studies in which a core concept is recording various physiological signals through a method called polysomnography (PSG). These signals include electroencephalogram (EEG), electrooculogram (EOG), and electromyogram (EMG), and are used to provide insight into how sleep changes throughout the night [3]. Conventionally, the signals are collected through electrodes wired directly to the body of the subject, which occasionally results in a situation where the subject has difficulties falling asleep. This, logically, is highly undesirable as it can affect the protocols in the study and yield expendable data. The project described in this document is the first step of a larger initiative aimed at creating a device capable of recording sufficient data from a patient in a way which is less intrusive of the subject’s comfort. To that end, this research will concentrate on the EEG signals only.

As of this moment, however, the analysis of the data collected from a PSG is done by displaying the signals parallel to each other and separating them into segments of a certain length, called epochs. Each of these epochs is then assigned to a sleep stage according to a set of rules given by either the” Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects” by Allan Rechtschaffen and Anthony Kales (R&K) or the AASM [4], [5]. As of the moment of writing this document, scoring the epochs is done visually, by trained experts. After all the data have been scored, a graph can be created where the evolution of a night’s sleep can be observed (hypnogram) and conclusions drawn about the experiment.

However, the method of visually scoring the data is inherently flawed. It allows for a certain level of freedom in scoring which ultimately results in conflicting scores by separate scorers. A study by the AASM reports 82.6% overall sleep stage agreement between 2 500 scorers, with the agreement falling to 63.0% for some sleep stages [6]. It is apparent that this inconsistency is a major point for improvement as it undermines an important cornerstone of science, namely repeatability. Additionally, it is easy to imagine how visually assimilating hours of recordings across multiple subjects requires a lot of time. We believe that an automated system will address these challenges.

The creation of an automatic sleep scoring system is not a novel idea and in recent years several research groups have proposed their own methods to automate sleep classification [7]–[12]. These methods make use of various signal processing techniques paired with machine learning (ML) algorithms to determine the stages of sleep from PSG data. The signal processing techniques are aimed at extracting distinct characteristics (features) from PSG signals which can then be fed into a classifier algorithm in order to “teach” the program how to distinguish between the sleep stages.

Most of the previously done studies concentrate on supervised techniques for learning, where the algorithms learn by fitting data that has already been scored by a human to create its decision-making rules. This means that the algorithm will attempt to create rules which simulate the decision-making of the scorer [13]. It can be argued that using supervised learning algorithms, in this case, is not the best option because of the inter-scorer agreement percentage mentioned earlier. The argument to be made is that even if the algorithm is 100% accurate, which is unlikely, it is only as accurate as the scorer used as its reference, which according to the numbers above leaves 17.4% chance of it disagreeing with other scorers. This is why training an unsupervised algorithm such as a Hidden Markov Model (HMM) [14] presents an interesting point for discussion.

Thus, this thesis will aim to answer the following research question:  
What are the requirements for an automatic sleep stage classification based on single channel EEG signal data with regard to its accuracy, sensitivity, and specificity?

* Which features are yielding the highest segregation between the classes?
* Which EEG channel yields the best performance?
* Which ML technique yields the highest accuracy, sensitivity, and specificity?
  + Are there any additional techniques which improve the performance? (feature selection, dimensionality reduction, statistical tests, data manipulation)
* Is the HMM a suitable approximation of the cyclic nature of sleep and does the Viterbi algorithm yield better results than other ML techniques?

CHAPTER II

SITUATIONAL & THEORETICAL ANALYSIS



2.1 Sleep and sleep studies

2.1.1 Effects of sleep on the human body and behavior

As mentioned in the first chapter, there is extensive proof of the negative effects of sleep deprivation in humans. It is perhaps intuitive to think that the effects of sleep deprivation are most apparent on the neurobehavioral level. It has been shown how sleep restriction leads to lapses of attention, depression, preservation and continuation of thought, slowed working memory, and generally reduced cognitive performance [15]–[17]. Moreover, these deficits accumulate and could result in micro-sleep episodes and daytime drowsiness[15], [18]. In practice, this is not only reducing overall performance during the day but could potentially have critical consequences for people who operate heavy machinery such as long-distance truck drivers [19].

Even though the cognitive effects are immediately apparent, sleep restriction also affects the human physiology. Data from epidemiological studies find a strong correlation between sleep deprivation and an increased body mass index (BMI), suggesting that sleep restriction leads to obesity 64, 65. Additionally, there is evidence showing reduced leptin levels [20] and reduced tolerance to glucose [21] in sleep-deprived individuals. Moreover, the review reports increased blood pressure [22], changes in the activation of the sympathetic nervous system [23], and increased levels of inflammatory markers [24]are showing the effects of sleep deprivation not only on the endocrine system but these on the cardiovascular and immune systems as well.

Unfortunately, the studies summarized in the review do not provide enough information to conclude what the function of sleep is or describe its underlying principles [2]. It would seem however that the function of sleep is not restricted to simply avoiding sleep deprivation, as recently it there has been a hypothesized link between sleep and memory and learning [25]. In any case, sleep remains somewhat of a mystery and therefore more research must be done to comprehend it.

2.1.2 EEG signal and the R&K and AASM systems of classification

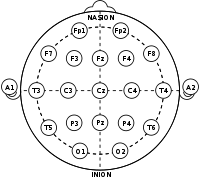
An encephalogram (EEG) is a method of monitoring the electrical activity of the brain. In most cases, it is a non-invasive procedure utilizing 21 electrodes placed according to the international 10-20 system shown in figure 1 but higher resolution systems are also possible utilizing a higher number of electrodes. The EEG signals are extracted from the difference in electric potential between two electrodes. Therefore, the signals are referred to as channels and each channel is given by the two electrodes from which the potential has been measured. Using all of the 21 electrodes is unlikely during sleep studies. It is much more common to use a single channel EEG, meaning only a few electrodes are attached. This is done both because of the whole configuration of 21 is uncomfortable and also because a single channel might prove sufficient when it comes to sleep stage classification given that the full EEG presents redundant information [5]. Additionally, the possibility to have only two electrodes has seen the rise of multiple wearable devices such as headbands or earpieces that could record the single channel EEG.

Figure 1 The 10-20 system for EEG electrode placement

The electrical activity of the brain during sleep has been studied extensively and its easily distinguishable patterns have been described [5]. As of this moment, 2 separate but similar standards of scoring sleep EEG signals exist: R&K and AASM as shown in table 1.

Table 1 Comparison of sleep stage classification standards [4], [5]

|  |  |
| --- | --- |
| System | Sleep stages |
| R&K | REM, S1, S2, S3, S4, Awake |
| AASM | REM, N1, N2, SWS, Awake |

The sleep stages of both systems are characterized by the waves in an individual epoch. There are 6 distinct types of waves and events described below and in figures 2 and 3.

* Delta waves (up to 4 Hz with the highest relative amplitude)
* Theta waves (4 Hz-8 Hz)
* Alpha waves (8 Hz – 14 Hz)
* Beta waves (14 Hz – 30 Hz)
* Gamma waves (30 Hz – 100 Hz)
* Sleep spindles and K-complexes (individual artifacts)

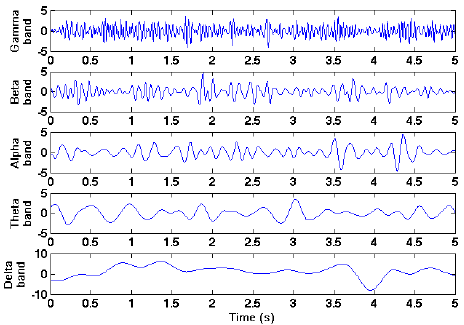


Figure 2. Frequency bands of the sleep EEG signal represented as signals in time.[26]

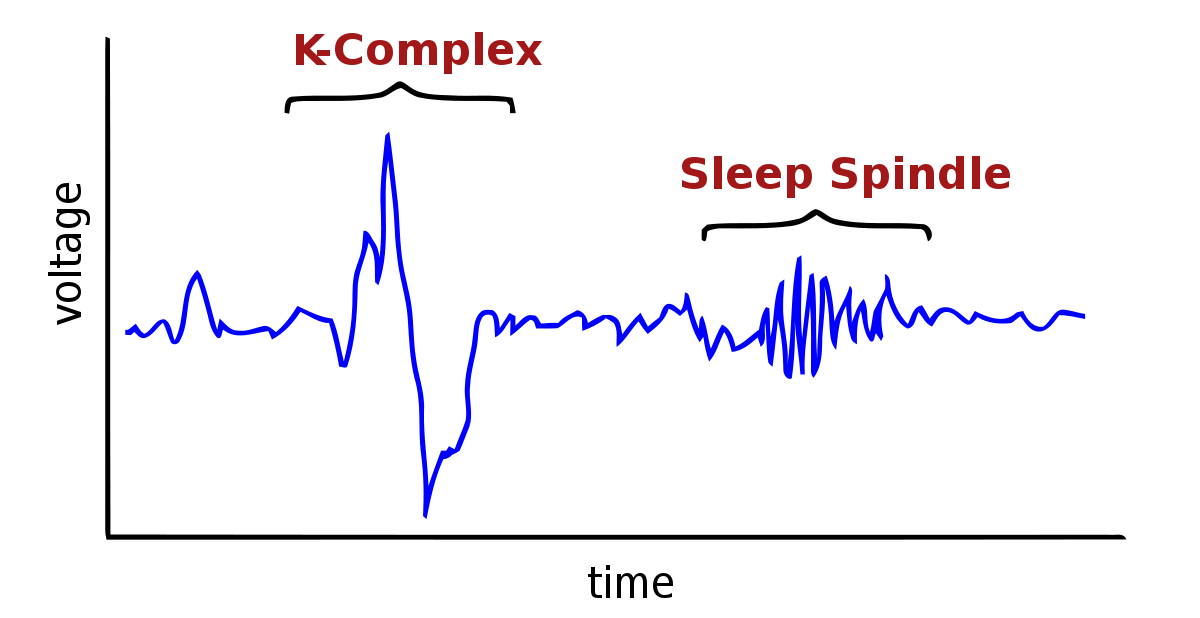


Figure 3. K-complexes and Sleep spindles as parts of an EEG signal in time

Stage 1 sleep is often described as drowsiness. It is a stage of light sleep and some alertness still remains. It is characterized by alpha and theta waves [5]. It is typically a short period of typically up to 7 minutes [27].

Stage 2 is similar to stage 1 in the meaning that the sleep is still fairly light. However, in this stage, the brain produces sudden changes in brain activity. These are the previously mentioned sleep spindles. Their presence is technically enough to classify an epoch as a stage 2 sleep. However, it is worth mentioning that k-complexes also occur in this stage of sleep [5].

Stage 3 marks the beginning of deep sleep and the brain produces slow delta waves. In this stage, the body does not move and much less responsive to outside stimuli. It is characterized by 20-50% of delta waves in an epoch [5].

Stage 4 is the stage of deepest sleep. This is the stage at which it is most difficult to wake up a person [5]. Stage 3 and 4 are known to represent up to 25% of sleep in children and drop to 10% by the age of 60 [27]. It is characterized by more than 50% of delta waves in an epoch [28].

Rapid Eye Movement (REM) sleep is, as the name suggests, characterized by rapid eye movements. REM sleep episodes become longer as the night progresses. REM sleep is thought to be the stage at which dreaming occurs. The heart rate is irregular, the breathing is irregular and there are bursts of muscular twitching [5].

2.1.3 Circadian rhythm and Sleep homeostat

Sleep characteristics such as timing, structure, and propensity, are dependent on two major factors: the circadian rhythm and the sleep-wake homeostasis. The circadian rhythm is the innate internal ~24-hour long cycle which is regulated by the amount of light and thus is parallel to the day-night cycle under normal conditions. Studies have shown how the circadian pacemaker, located at the suprachiasmatic nucleus (SCN), controls the drive for sleep by regulating the bodily functions through the release of hormones. The sleep-wake homeostasis, on the other hand, adds to the drive for sleep based on how much sleep the subject has had a priori. The interaction between these two factors is complex and each of them influences the separate stages of sleep in a different way [29].

In any case, it can be concluded that sleep is a time-dependent process. In fact, it has been observed that an oscillation exists between the separate stages of sleep. This oscillation is illustrated in figure 4 in which the separate cycles of sleep are shown as a function of time during an 8-hour long night sleep.

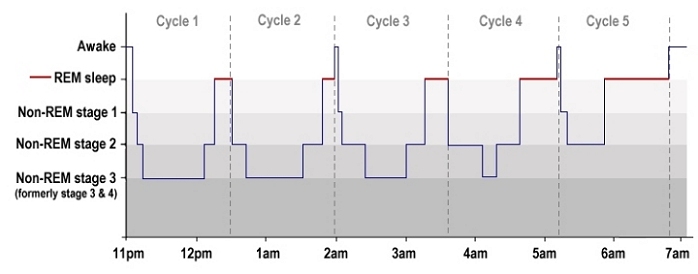


Figure 4. A hypnogram showing sleep stages and cycles in time [30]

2.2 Machine Learning

2.2.1 Overview

Machine Learning can be summarized as a set of techniques which are enabling a computer program to “learn” without an operator explicitly casting commands. This practically results in a program which can make predictions about data it is not familiar with by applying statistical methods. A common task for ML algorithms is discrimination between clusters of data. In order to do that an algorithm must be able to formulate its working space in the n-dimensional space and then draw borders between the data points such that clusters are formed. Naturally, the best performance would be achieved when the data points can be spread out in space such that easily distinguishable clusters are formed. This can be achieved by looking at the data from different perspectives and through different data features. The features having the highest variance between data points are the ones which will yield the best performance of any ML algorithm as shown in figure 5 [31]. Therefore, feature extraction is a critical step in building any automatic classification system.

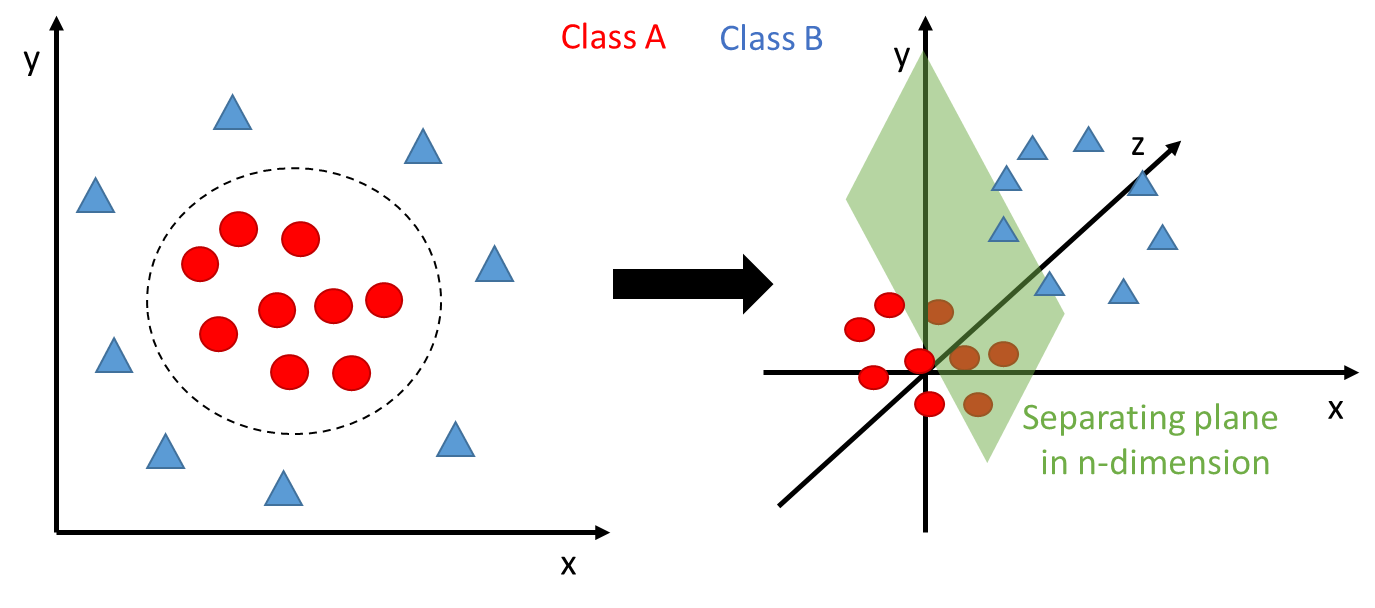
****

Figure 5. Data visualisation under different feature spaces. Moving from a lower dimension feature space (left) to a higher dimension feature space (right) can make it easier to segregate the classes.

2.2.2 Methods for feature extraction

2.2.2.1 Time-domain

**Time-series signals**

It is clear from part 2.1.2 that the data we use is a time-series, meaning a signal composed of data points recorded in time. As we mentioned the data is scored by humans and the scores given by the human experts are a single value attributed to an epoch. These epochs are of finite length, typically between 4 and 30 seconds, which means that they are represented by a given number of data points depending on the sampling frequency of the data [4], [5]. Since the data is typically sampled at more than a 100Hz it becomes obvious that the length of an epoch will most likely exceed 400 data points. If we feed this raw EEG epoch data to an algorithm, the hyperplane in which it operates will have 400 dimensions. This is important because in many cases the computational time of the algorithm depends on the number of dimensions. While 400 dimensions might not be considered a large number for Big Data applications we should keep in mind that this is calculated by taking the minimum mentioned requirements. In fact, in our case the number of dimensions for the lowest number of data points is given by the 5 second epochs sampled at 200Hz. Additionally, simply feeding the raw data in an algorithm is perhaps not the best option because each dimension is represented by the data point at a given time step. This leads to the conclusion that representing an epoch with as few numerical values as possible is beneficial. The easiest way to do this is by extracting the statistical moments of a time series.

**Statistical moments**

A common step in analyzing EEG signals is extracting their statistical properties such as mean, standard deviation, skewness [10], [12] given by the following formulas:

where **x** is a data point and **N** is the number of data points in an epoch.

Even though we have defined the formulas it is best if we graphically show what they represent in order to understand why we hope they are a suitable representation of the epochs. Figure 6 represents an example of all four concepts in succession. As we can see all show distinctive characteristic of the population, meaning the data points in the epoch.

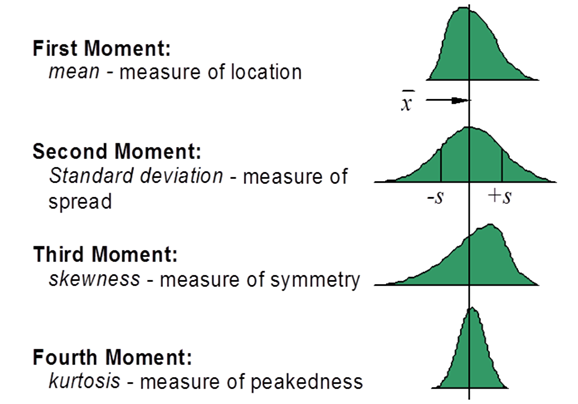


Figure 6. First 4 statistical moments visualization and explanation

**Zero-crossing rate**

Another simple calculation that can be done in the time domain of a time series is how many times the signal crosses the zero on the x-axis [10]. This measure, called zero-crossing rate, is also given by how many times the signal has changed its sign. It is determined by the following formula and displayed in figure 7:

Where x is the signal of length T and is an indicator function given by:

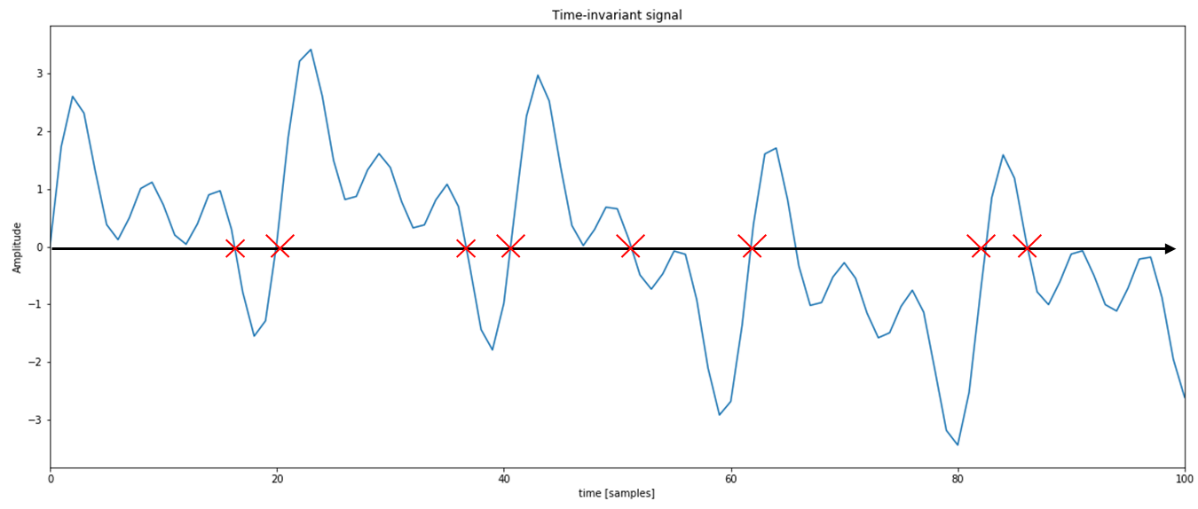


Figure 7. The zero-crossing rate of a time-invariant signal. Notice how the number of zero-crossings is not representative of the frequency of the signal

**Hjorth parameters 48**

During the early seventies, Hjorth introduced his parameters which describe the seemingly most basic signal properties. These descriptors are called activity, mobility, and complexity and are given by the following formulas [32]:

Where var is given by:

Where μ is the mean.

These parameters are very intuitive for signal characterization even if the formulas might suggest otherwise. The first parameter, Hjorth activity, is given by the variance of the signal and essentially is a measure of the mean power of the signal. The Hjorth mobility is a measure of the mean frequency. Practically, mobility could be interpreted to yield the dominating frequency. From equation 10 we see that complexity is heavily based on mobility. It basically yields the bandwidth of the signal [32].

2.2.2.2 Frequency domain

**Representation in the frequency domain**

We already touched on the topic of extracting the frequencies from the time series. However, none of the methods mentioned in section 2.2.2.1 yields the exact frequencies present in the signal. The most common method for extracting the frequencies out of a signal is the Fourier Transform (FT). The FT dictates that any waveform can be decomposed to its fundamental sinusoidal functions. Therefore, what the FT gives us is the intensity of every frequency present in the time series. Since our signal is a time-series with certain sampling frequency we would need to use the discrete Fourier transform (DFT)[33] given by the following formula:

Where is the transformed data point at location ***k***, ***N*** is the number of data points and is the current sample at location ***n***.

This transform works under the assumption that the signal is stationary, which would mean that it repeats infinitely with the same period. An additional assumption is that the signal which is fed into the transform is of large enough sample size to be able to capture at least one period of the components which make it. These assumptions are relevant because when the transform is applied to a time series which does not conform to the requirements the results are poor [33].

If the signal is not stationary, the DFT magnitude plot would yield information which might not be particularly useful for our purpose as seen in figure 8. The situation there happens because the frequency representation has no temporal information as it gives all the frequency intensities at all times in the signal [33].

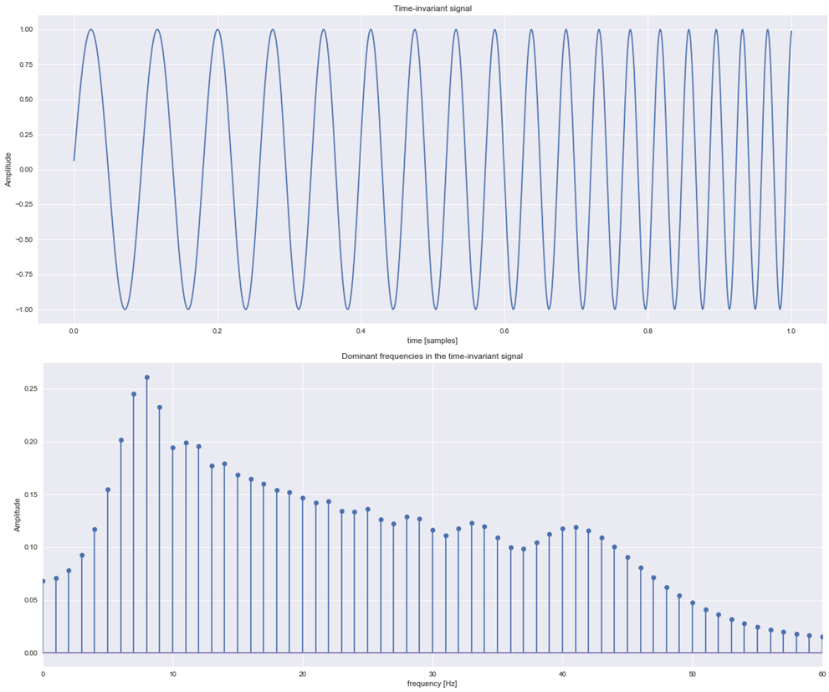


Figure 8 A time varying signal and its DFT magnitude plot.

If the sample size is not representative the DFT would yield a magnitude plot with an overwhelmingly dominant 0th frequency. This happens because, even though the sinusoid composing the signal might be infinity, the DFT takes only the presented part in its window and so results in a dominant zero frequency [33] as seen in figure 9.

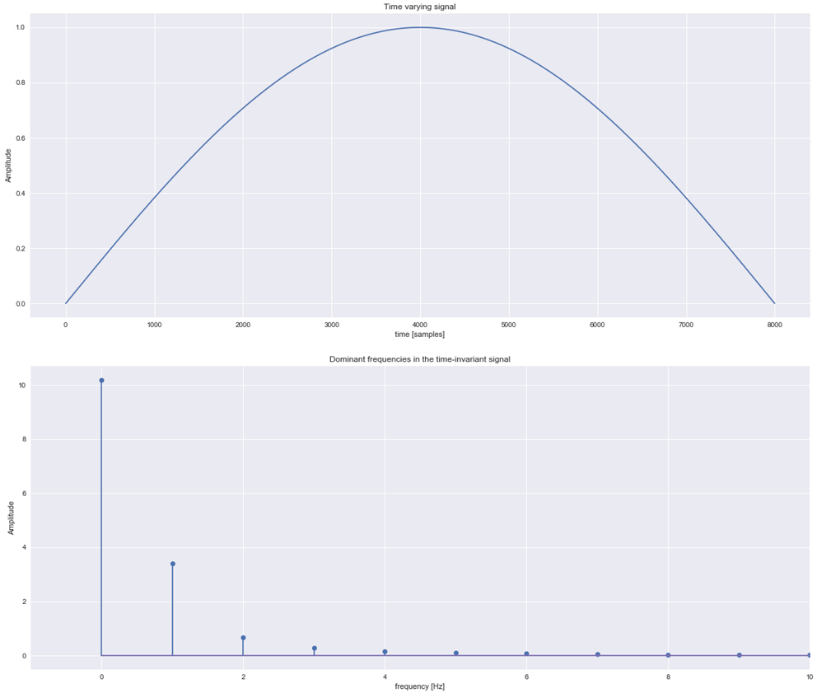


Figure 9 Half a sine wave and its DFT magnitude plot.

In our scope, this is important because whether sleep EEG experiences stationarity is debatable [34]. In any case, the frequency information which is contained in the epochs is largely relevant for our automated system. The manuals for human scoring heavily depend on frequency information to determine the sleep stage and therefore it makes sense to try to extract this information and feed it to our system [5]. So far, we have identified two methods on how to tackle this challenge, both of which are described in the following sections.

**Power Spectrum Density (Non-parametric)**

The first method of representing the frequency domain characteristics of a signal as features for an automated algorithm is through the power spectrum density (PSD) bands. Notice how we have included the word density in this case. This is because the sleep EEG signal experiences aperiodic behavior [34] vaguely reminiscent of the case displayed in figure 7. In these aperiodic cases, the power spectrum is defined as a continuous function rather than a discrete set of frequencies. In the case of a discrete set of frequencies (periodic signal), each frequency component has a unit of power. However, when the spectrum is given by a continuous function the units are power per frequency, which make sense from the conclusion we reached about the temporal invariance of the representation where all frequencies are given at all times [35]. In this situation, we have to integrate over an interval of frequencies (band) in order to get the power in that interval. Alternatively, we could extract the PSD, which gives us the weight a particular frequency carries in the overall power of the band, by calculating the autocorrelation between frequencies in the spectrum [35].

The first method we use is based on integrating a divided power spectrum of 5 frequency bands (Delta: {0.5-4Hz}, Theta: {4-8Hz}, Alpha: {8-14Hz}, Beta: {14-30Hz}, Gamma: {>30Hz}) which are also used to characterize the separate sleep stages. We use the values given by the integral as the features to characterize the epochs [5]. The process is visualized in figure 10.

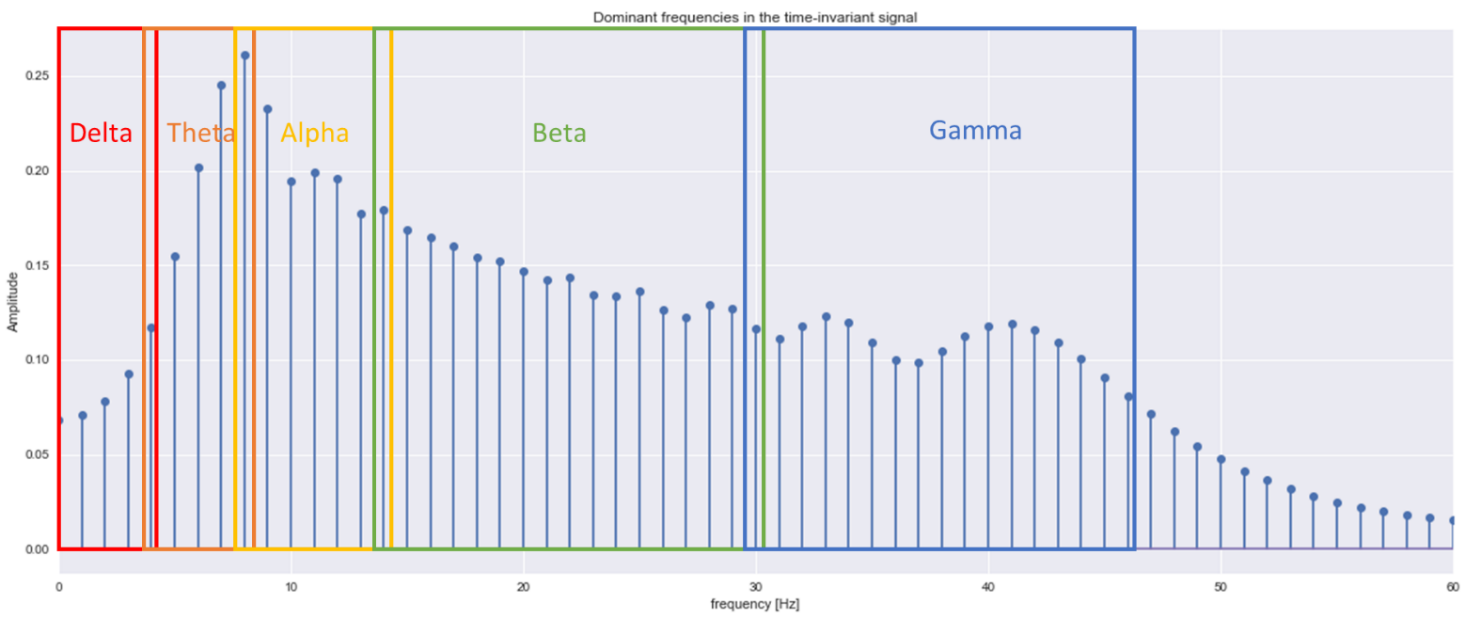


Figure 10. Frequency power spectrum of an aperiodic signal with colour coded spans of the frequency bands used for sleep classification.

**Autoregressive model coefficients (Parametric)**

The non-parametric method described above is typically used when there is little information about the nature of the signal. Going into a parametric method for extraction of the PSD requires some assumptions about the signal. Previous research suggests that an autoregressive (AR) model can be used to sufficiently approximate sleep EEG. Therefore, it is not unreasonable to assume that our signals can be modelled by an AR process [36], [37].

An AR process is a random process in which the output is assumed to depend linearly on its previous steps and on a stochastic term. This stochastic term is an imperfectly predictable term usually given by white noise. In essence, this means that we can approximate a quasi-random process by generating white noise and combining it with its weighted previous steps as given by the sum in equation 11 or by the series in equation 12 [37]:

Where an AR(p) is an autoregressive model of order **p,** are the coefficients (weights) of the model, is a constant and  ***εt***  is white noise. By looking at the formula, we can conclude that the variable which gives us the most information about the signal is the coefficients given by the Greek letter phi. In fact, this trend continues in the formula for power spectrum estimation of an AR process [37]:

Where is the variance of the white noise. Since everything besides the coefficients in this formula are either constants or have a trend to their behavior, we can use them as a characteristic for the PSD of the signal model. It is important to emphasize that the PSDs we extract from the epochs used in the non-parametric method from section 2.2.2.2 and the ones we simulate with the AR model should theoretically provide almost identical magnitude plots as seen in figure 11. Therefore, it will be redundant to repeat the step of separating the spectrum into bands and integrate the data. Instead, we will use the coefficients themselves as features for the automated scoring algorithm. It is reasonable to do that because we would not gain any new perspective on the data if we repeat the same process from the non-parametric part, but the coefficients present a new hyper plane from which to analyze the epochs. A visualization of what the coefficient values look like is given in figure 12.

Since we have decided to use the coefficients we would need a way to extract them from our own data and fortunately for us, the Yule-Walker equations allow us to easily do that. We can express the Yule-Walker equations in the compact matrix form [37]:

Where *R* is the correlation matrix containing the calculated values from the autocorrelation function of the AR process, *w* are the weights given by and *r* are the values for the correlation. Thus we can solve for *w* assuming that R is nonsingular (can be inverted) from the compact matrix form as follows [37]:

Which in expanded form would look as follows [37]:

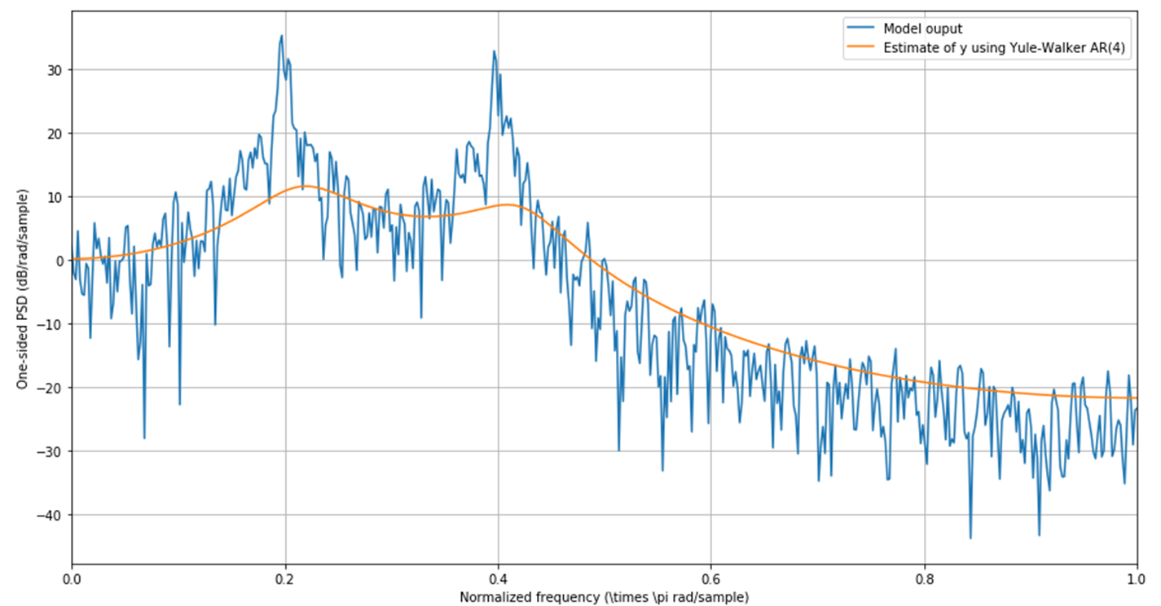


Figure 11. A generated AR model PSD output (blue) and its estimation based on the coefficients (n=4) extracted by using the Yule-Walker equations



Figure 12. Evolution of the AR parameters with increasing **i** term

2.2.2.3 Time-frequency domain

**Time-varying signals**

While the PSD representation provides us with an idea of the frequencies present in the signal it has two distinct drawbacks. As the standard frequency representation given by the DFT, it lacks any temporal information. However, more importantly, the PSD is a measure of frequency bands and not the frequencies themselves. We would perhaps gain greater accuracy from an ML algorithm using a representation which accounts for both of these issues, hence we require a signal representation in both the time and frequency domain simultaneously [33].

**The Heisenberg uncertainty principle**

Before discussing the time-frequency domain representation of our data it is also important to understand an inherent limitation that bounds these representations. One of the fundamental concepts in physics is the Heisenberg Uncertainty Principle, also known as simply the Uncertainty Principle (TUP), which dictates that it is impossible to measure both the exact velocity and the exact position of an object. TUP is inherent to the mechanics of waves so naturally, it carries over into the signal processing domain, where frequency representations are abundant. In fact, contrary to its name, TUP has nothing uncertain about it. For all practical purposes, it means that it is impossible to locate the exact frequency component at the exact time it occurs [33]. In our case, this forces us into deciding which we value more, the frequency information or the temporal one.

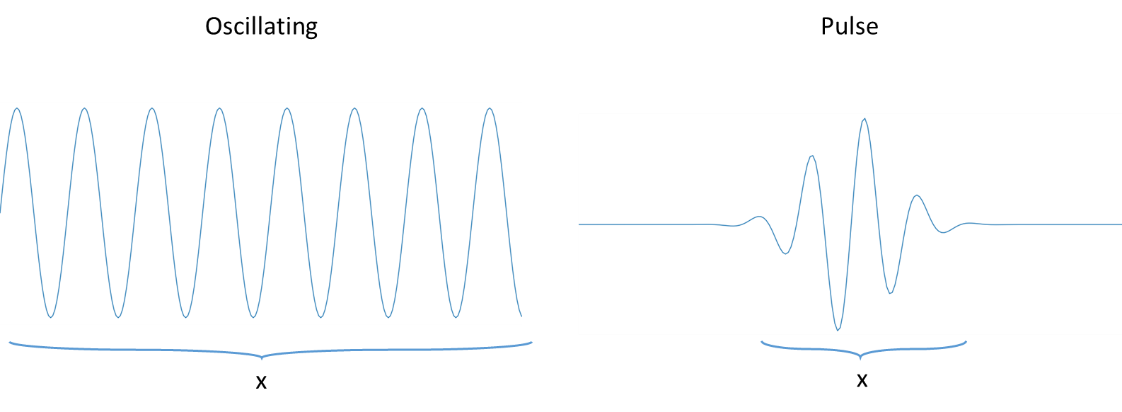


Figure 13. Visualization of the uncertainty principle. On the left, the oscillating wave presents more information about the frequency (momentum) of the wave (particle) and insufficient information about its position, and on the right, the pulse wave presents much more information about the position of the particle but less about the frequency.

**Short-Time Fourier transform (STFT)**

As already discussed, due to TUP we cannot extract the exact time-frequency information of a time-varying signal. Fortunately, in our case, it is possible to overcome this limitation to a certain extent by selecting a range of frequencies (band) to examine in a given time interval (window). Doing this we receive both time and frequency information simultaneously, even if it is not exact. The first technique which we use in this study is the discrete-time STFT defined as [36]:

Where x[n] is the signal and w[n] is a window function. What STFT essentially computes, is the Fourier Transform of the signal in the window space as shown in figure 14. In it a rectangular window is used to define limits to the part of the signal which is manipulated. Using a window however, presents multiple challenges such as selecting the type of the window used, number of samples in the window, overlapping of windows, zero-padding of the signal [36].

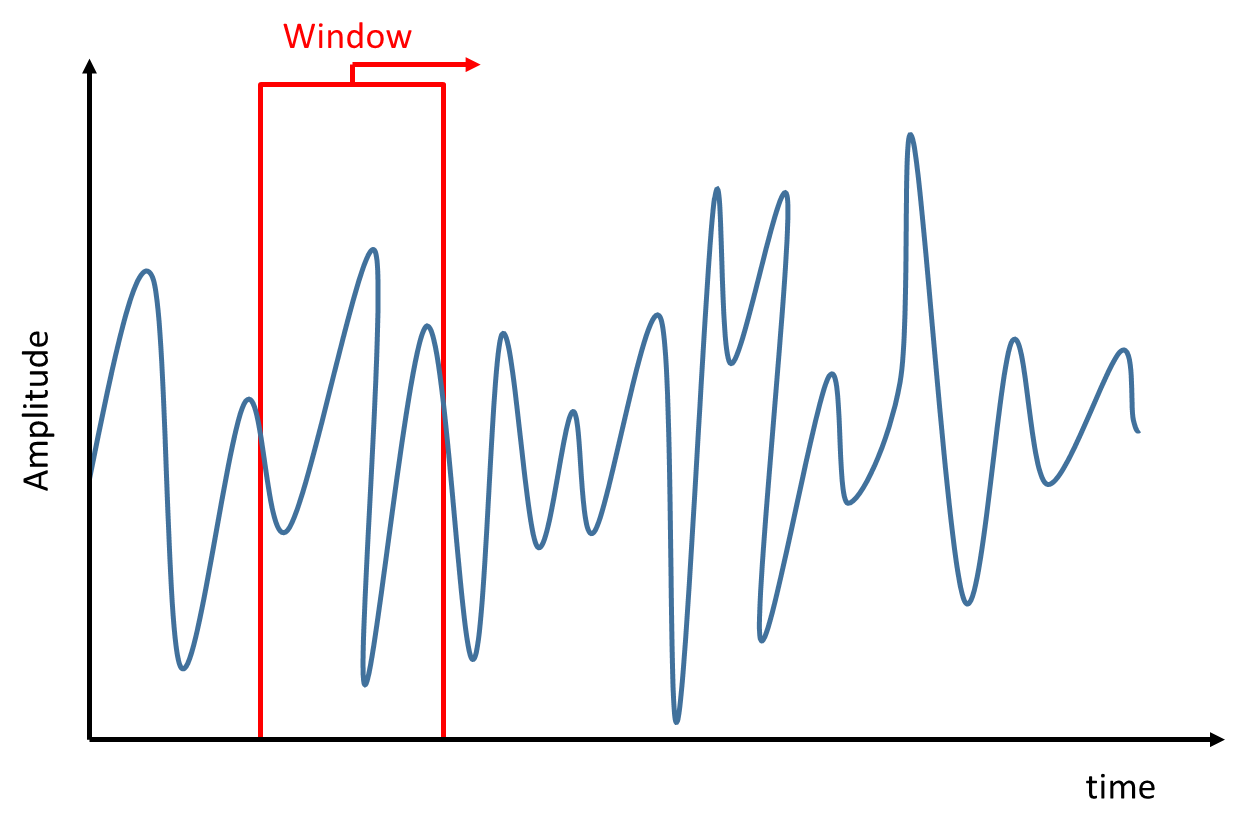


Figure 14. A representation of a rectangular window sliding over a time-series. In STFT this window defines the limits of the Fourier transform as the signal outside of the window is ignored.

The window function is an important source of variables for the STFT algorithm. It is a function which is zero-valued outside the window area and real-valued under it. In the case of figure 3 it is zero everywhere outside the square. However, windowing a waveform such as cos and taking its Fourier transform yields non-zero values for frequencies other than ω. This is called spectral leakage and is highly relevant in our case because we have both a low frequency resolution due to TUP and also multiple frequencies because of the nature of our signals. However, window types other than the rectangular exist and they all have different spectral representation [36]. Several of these windows can be seen in figure 15, where both the shape of the window in the time domain is shown in blue and the respective spectral representation shown in orange.

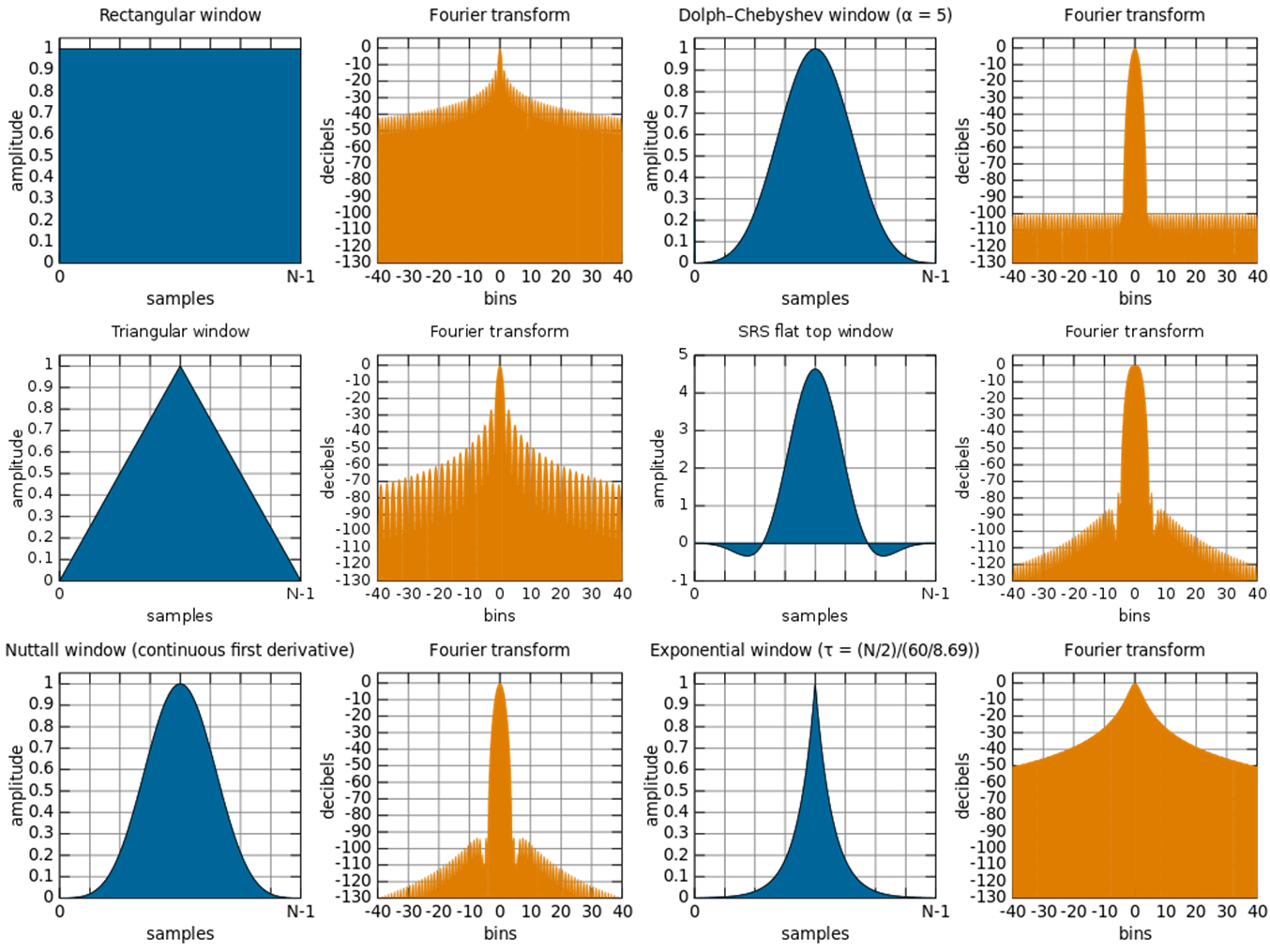


Figure 15. Different window functions represented in the area they cover in time-series representation (blue) and the Fourier transform of a waveform under the respective window function (orange)

As it is clearly seen in the figure, opting for a Nuttall window or a Dolph-Chebyshev window will decrease the spectral leakage. However, the representation of the window in the time domain creates a new challenge. If the Nuttall window is chosen for example, some of the time-series signal might not be covered by the area of the window as illustrated on the left in figure 16. There a significant part of the time-series is left outside of the window and thus not included in the Fourier transform. This is of course undesirable as not taking the signal in its entirety means that the spectral representation would not give us the correct information. A common technique used to overcome this issue is overlapping the windows as seen on the right of figure 16. This adds a new dimension for consideration when implementing the algorithm in the form of a compromise between the maximum amplitude “flatness” of the data weighting, shown in green on figure 15, and the overlap correlation of the spectrum estimation which would increase the computational effort unnecessarily if chosen to be too high [36].

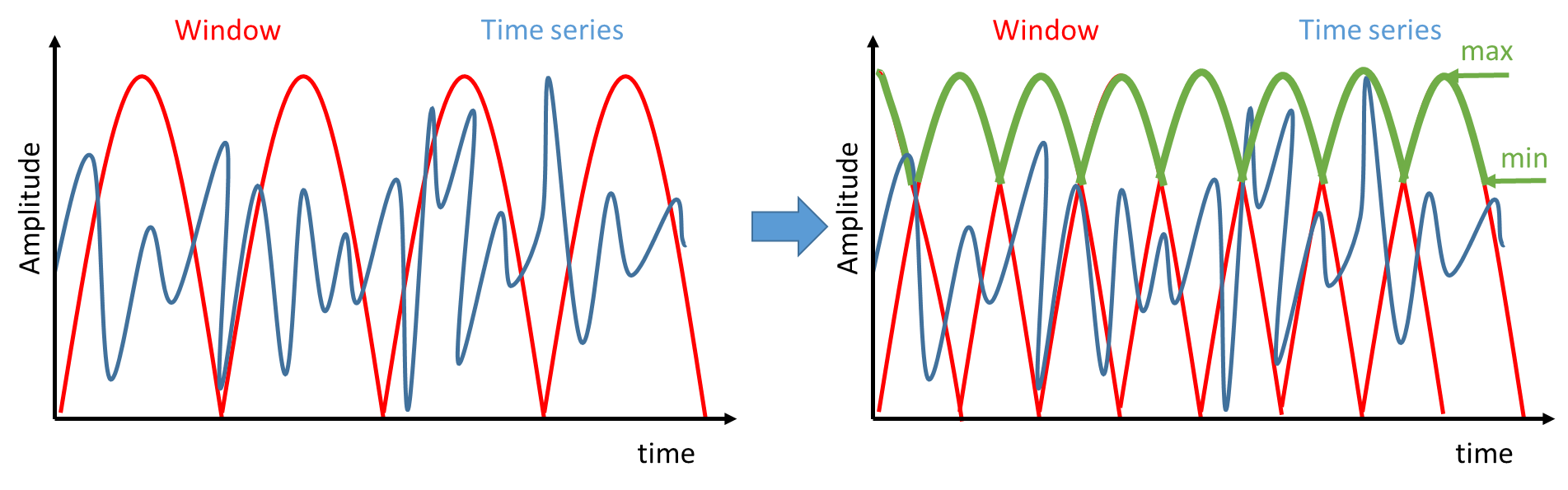


Figure 16. A Nuttall window (red) over a time-series (blue) without overlapping (left) shows how much of the signal remains in the zero-valued are (outside of the window). On the right it is shown how overlapping the windows resolves this issue as almost all of the signal is under the area covered by the windows (50% overlap) with the amplitude flatness shown in green..

Additionally, selecting how many samples of the signal must fall under the window (window span) is the variable that decides the frequency and temporal resolutions [36]. This parameter is the one that is directly related to the previously described TUP. The higher the frequency resolution, the larger the window, the least temporal information [33]. We can see this from a simple experiment where we use a time-varying signal similar to the one from figure 8 and apply the STFT with different sized windows. Figure 17 displays the results from this experiment for 4 differently sized windows (N represents the number of samples falling under the window, with 512 being the length of each of the distinct 4 segments (5Hz, 10hz, 15Hz, 20Hz)). In the top-left image where the length is the lowest, we can see that there is barely any overlap of the 4 segments and that the color, representing the power of the spectrum, is at its highest contrast. Once we start to increase the size of the window, the overlap starts to increase and the contrast to decrease. However, the frequency information becomes less spread out as seen from the decrease in “shadows” above and below the bars.

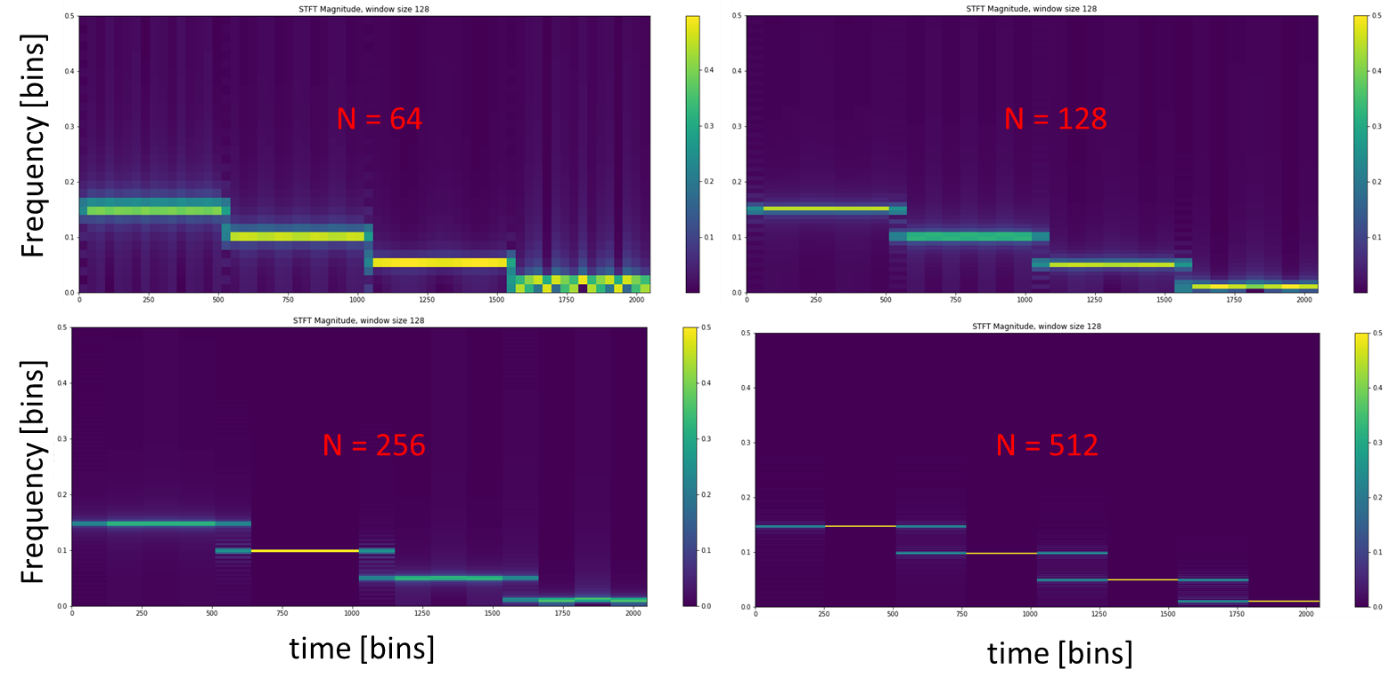


Figure 17. Spectrogram of the STFT of a time-varying signal with 4 distinct segments (5Hz, 10hz, 15Hz, 20Hz) of equal length (n=512) with varying window length (red N annotation). We can see how increasing the number of samples of a window decreases the temporal information (seen by the overlap of the segments and the decrease of colour contrast).

**Continuous Wavelet Transform**

A quick look at figure 16 clearly shows that the dominant frequencies are at the bottom of the plots. This is because the Nyquist frequency is the highest one on the plot, which in this case would be 64 Hz (128 Hz sampling frequency). However, we know that the frequency bands which are indicative of the sleep stages are at the lower end of the spectrum. The STFT provides for linear representation of the weight of each of these frequencies which is unfortunate because having a higher resolution at the lower frequencies might prove better for the performance of an ML algorithm. A technique called the Continuous Wavelet Transform (CWT) allows us to do something similar [33].

The CWT is a technique for time-frequency analysis which ultimately compares two signals: the original signal to be analyzed and one generated by us called a mother wavelet. This mother wavelet can be stretched or compressed and it can also be shifted in time. This stretching and compressing, given by the scale variable, allows us to gain information about the frequency of the signal. A tightly compressed mother wavelet would have higher similarity to the high-frequency parts of the signal while a stretched wavelet would be correlated to slower oscillations. By scaling the mother wavelet multiple times and shifting it throughout the whole original signal we can create a 2D representation of our signal where one dimension will be given by the scale of the wavelet and the other by the time shifts [33]. The process is illustrated in figure 18.

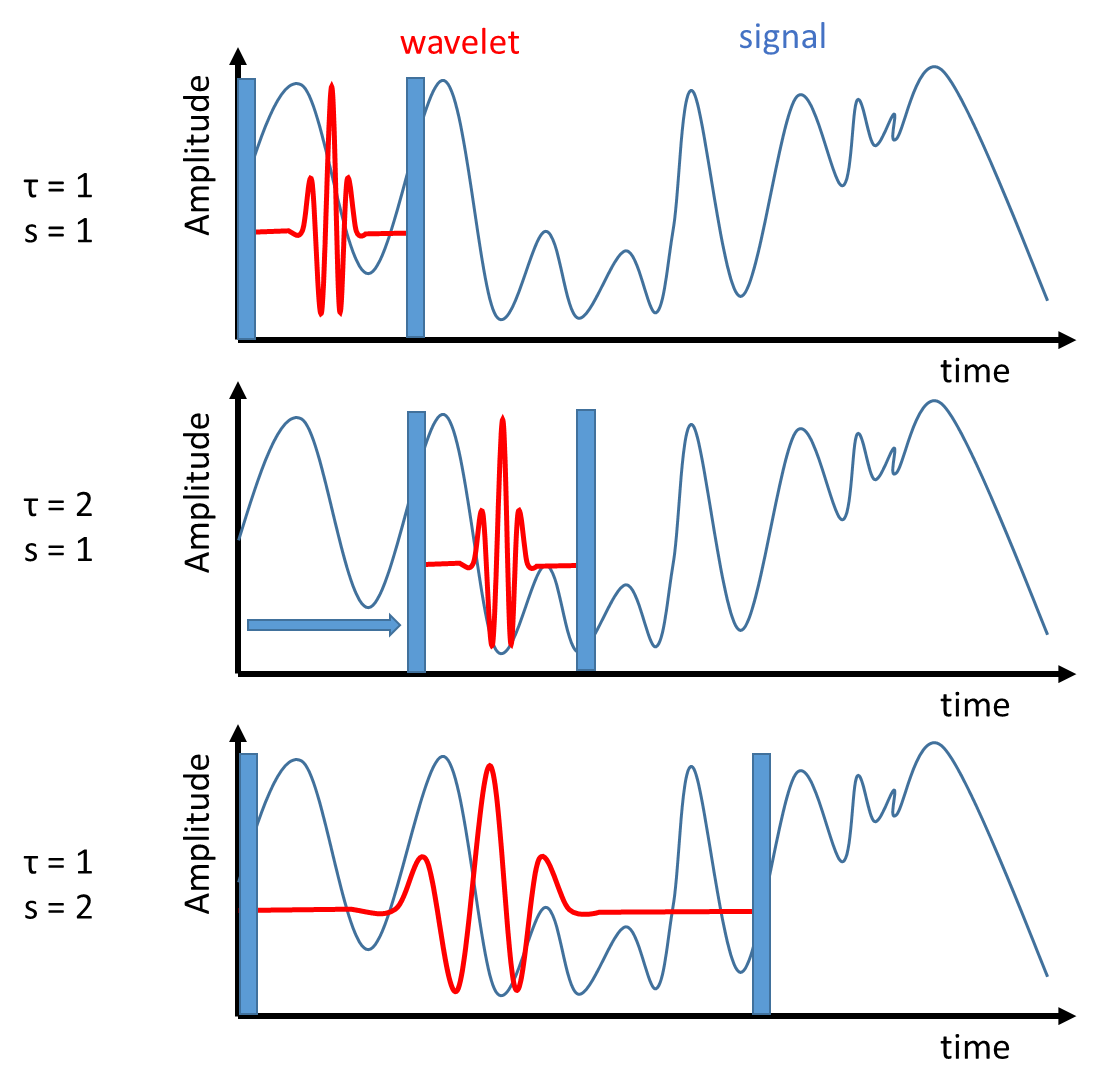


Figure 18. A visual representation of how the CWT algorithm works where **t** indicates time step and **s** indicates scale. First a low scale is selected (compressed mother wavelet) and it is being time-shifted as to be compared to every temporal bin of the signal. Afterward a new scale is selected and the process repeated.

The CWT is given by the following formula [33]:

where ϕ is the complex conjugate of the mother wavelet signal represented in both the time and frequency domain. As we can see the CWT does not exactly yield the time-frequency representation but rather variables whose values give the same information. On top of that, the CWT analysis allows for a much better time and frequency localization as we can have a non-linear representation as seen in figure 19. While the STFT gives us bins of equal width and length, the ones from the CWT are distributed unevenly which is exactly the characteristic we were hoping to achieve [33].

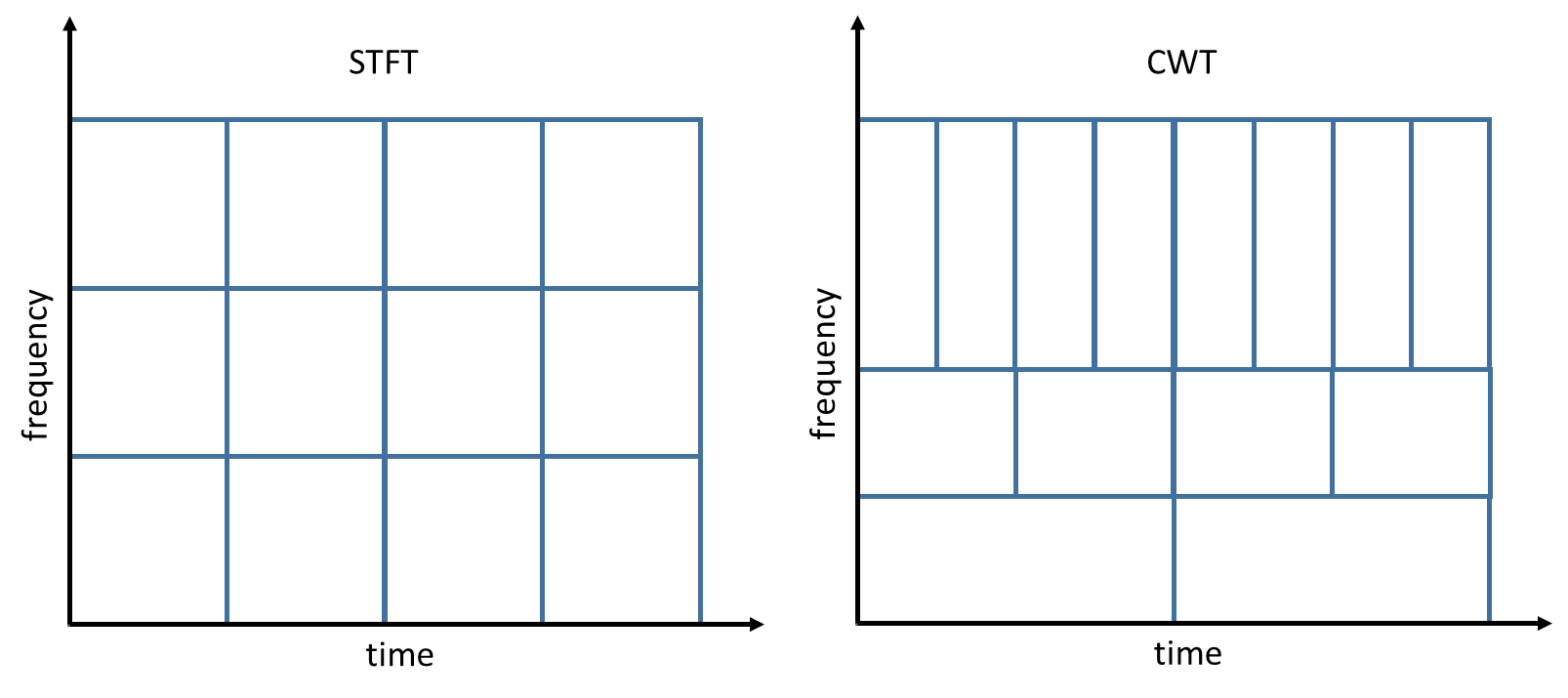


Figure 19. An example bin representation of the spectrograms of STFT (left) and CWT3 (right). It is apparent how the STFT has a linear distribution while the CWT allows for higher resolution at specific areas

2.2.2.4 Non-linear

Another type of features that can be extracted from a time-series is the non-linear subset. This includes measures such as sample and multiscale entropy, fractal dimensions, and chaotic attractor exponents [8], [10]. All of these are computed directly on a time-series without any preprocessing of the signal, yet they have a relatively high computational cost. Most of them are based on self-similarity computations which require significant working memory [38].

**Entropy measures**

Entropy measures provide an insight into the randomness or unpredictability of a temporal signal. The simplest algorithm is the one for the Approximate Entropy (ApEn). It yields useful information when the simple statistical measures of mean or variance would not distinguish between two series. An example of such a case would be when a series is composed of only two values one of which has a clear pattern to the occurrence of the values while the values in the other are distributed randomly [38].

The ApEn algorithm involves creating a sequence of vectors by sliding a window over the series where one of the requirements is to at least have the last value of the first vector overlap with the first value of the second vector. Then the ‘distance’ between each of the values between all vectors is calculated and compared to a chosen threshold value. In the case of ApEn, the values in the second vector must not be out of range (given by the threshold value) with the values of the second vector. Comparing all vectors to each other creates a new vector containing the number of times in which the condition has been satisfied. Then following the formula [38]:

where ***r*** is the threshold value, ***N*** is the number of data points, ***m*** is the length of the vector made by the sliding window, we calculate a value indicative of the unpredictability for this sequence. After this calculation, the process is repeated for ***m+1***. Finally, the ApEn is calculated by [38]:

**Fractal Dimension**

Conventionally, dimension is understood to be a measure of space. A unit in one-dimensional space would be a line, in two-dimensional space a rectangle, and in three-dimensional space a cube. Additionally, objects scale very well in the traditional sense of dimensionality as seen from figure 20. A separation of the 1D line in 2 translates into 4 rectangles in 2D and 8 cubes in 3D and so on.

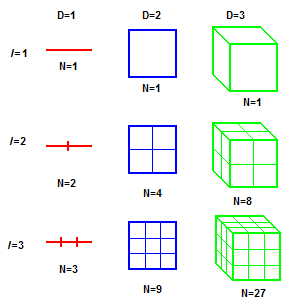


Figure 20. The conventional view of dimensionality and its scaling characteristic where D is the number of dimensions, I the number of initial separations, and N the number of resulting separations.

In fact, an equation can be derived from this relationship:

Where ***N*** is the separations number, the scaling factor, and ***D*** the dimension with representing proportionality. We can rearrange the equation as to make it yield the dimension instead [39], [40]:

This same rule will also be applicable in the case of fractal geometry. Fractals are abstract objects created by applying the original pattern of the object to each of its individual components, thus creating what is known as expanding symmetry. A typical example of this is the Koch snowflake shown in figure 21.

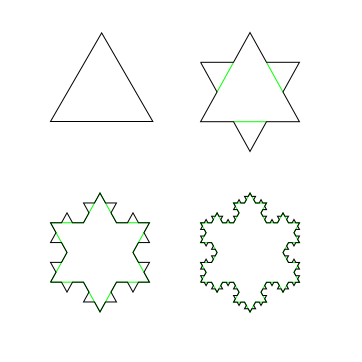


Figure 21. The first four iterations of the Koch snowflake, a typical example of a fractal.

In order to calculate the fractal dimension of the last instance of the Koch snowflake, we can assign the following values: N = 4 coming from the four iterations, and  **=** 1/3 coming from the three original sides of the triangle. Putting those numbers in the formula we find out that the fractal dimension of the Koch snowflake is 1.2619.

This simple example only illustrates the logic behind the fractal dimension, while in practice there are multiple techniques used to calculate it. Fractals model some natural patterns well up to a certain level. Self-repeating patterns are not uncommon in physiological signals such as the EEG either and therefore the fractal dimension of the signal might prove to be a suitable feature [40].

2.2.3 Classifier types

**SVM**

A Support Vector Machine (SVM) is a supervised ML model. It requires data to be represented as points in space in order to be able to separate them into categories. Typically, it is a linear model but it can be extended into the multidimensional space by using the kernel trick. After the data have been separated into clusters and classified, new data can be mapped onto these previously formed classes. This is called Support Vector Clustering (SVC) [41]. SVM attempts to create a plane that separates the classes by extending the margin of error between classes as seen from figure 22.

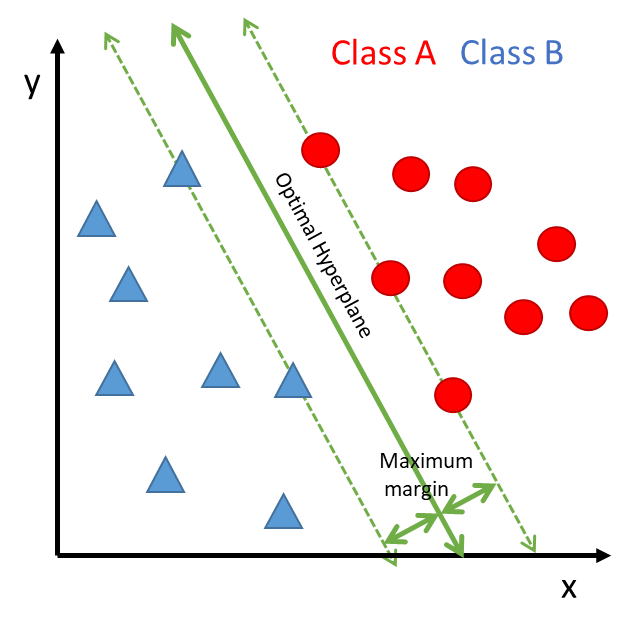


Figure 22. Support Vector Machine classification by creating a separating hyperplane in the feature space.

**KNN**

The K-nearest neighbor (KNN) method also requires a feature space in which to operate and classifies the data point based on a majority vote from its neighbors. The difference between the KNN and SVM is that SVM aims to minimize the error risk by maximizing the margin between classes and therefore creating a hyperplane separating them, while KNN is instance based and the class determined locally [42], [43]. The is visualized in figure 23 where we can see how changing the number of neighbors ***k*** might result in a different majority vote and therefore class assignment.

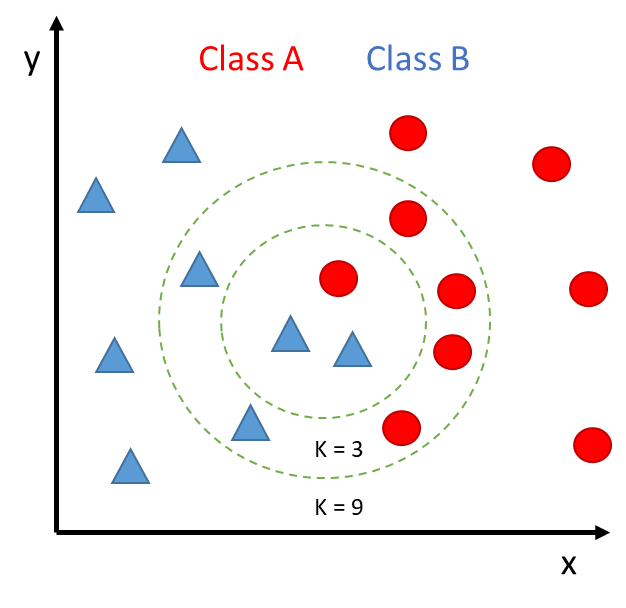


Figure 23. KNN algorithm under varying values k fro number of neighbors and the consequences of these changes

**DT/RF**

Random Forest (RF) is an algorithm which constructs multiple decision trees (DT) in order to classify data. These decision trees have conditionals which examine all the variance in the features to learn a model. Afterward, new data simply follows the conditionals until it is assigned a class [44]. The workings can be seen in figure 24.

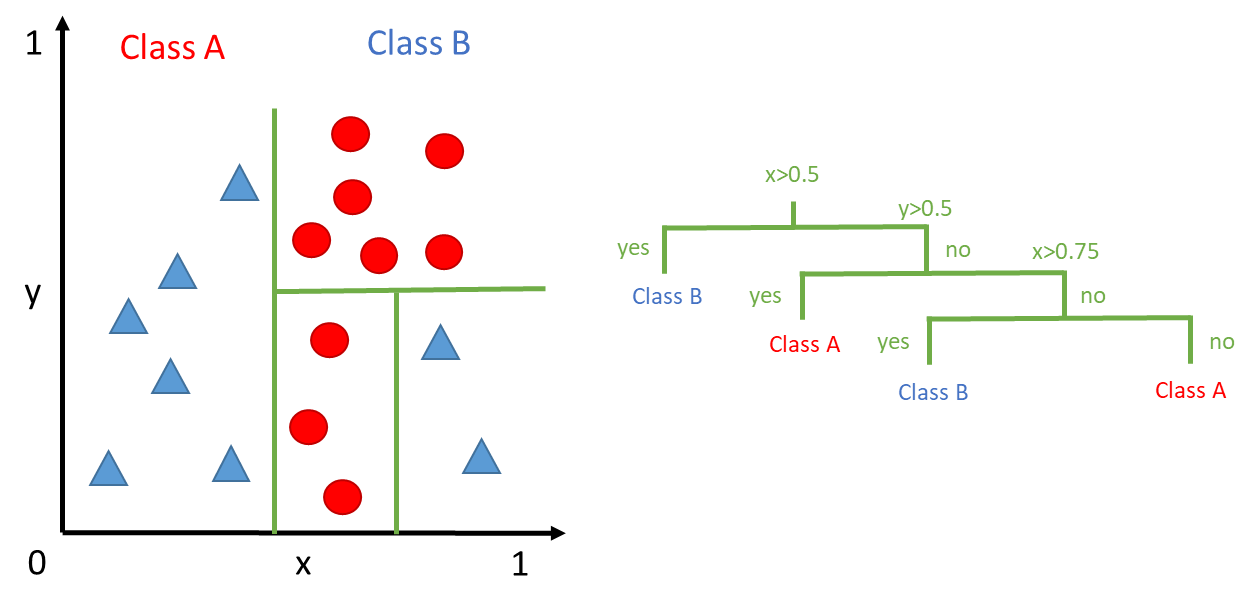


Figure 24. Decision tree schematic where the circles represent a single binary condition.

**HMM**

All of the previous ML techniques are supervised but there might be a benefit to trying to find structure in the data through an unsupervised algorithm. One way of doing this is by estimating a statistical model to an unknown system by observing its output. A statistical model is appropriate in this case because we assume that the EEG signal is the result of a stochastic process [34], [37]. In our case, we have a finite set of observations (EEG epochs) and we can assume that we have a random state switching process with the states being the sleep stages. Fortunately, these are all the preconditions required to train a Hidden Markov Model. But before we dive into how we can train an HMM from our data it is appropriate to examine the theory behind Markov chains first.

Markov chains are useful when we have a system which can be described as being in one of several states at any time. A typical example is a simplified version of weather observation, where we assume that the weather can be either sunny, cloudy or rainy presented in figure 25. There we can also see the probabilities of switching from one state to another given in matrix A. Then we calculate the possibility of having a certain sequence of observations occur. If we would like to calculate the probability of the sequence: Sunny, Cloudy, Cloudy, Rainy, we would only need to multiply the switching probabilities and assign a starting probability. If we are certain that the sequence starts with Sunny then this results in [14]:

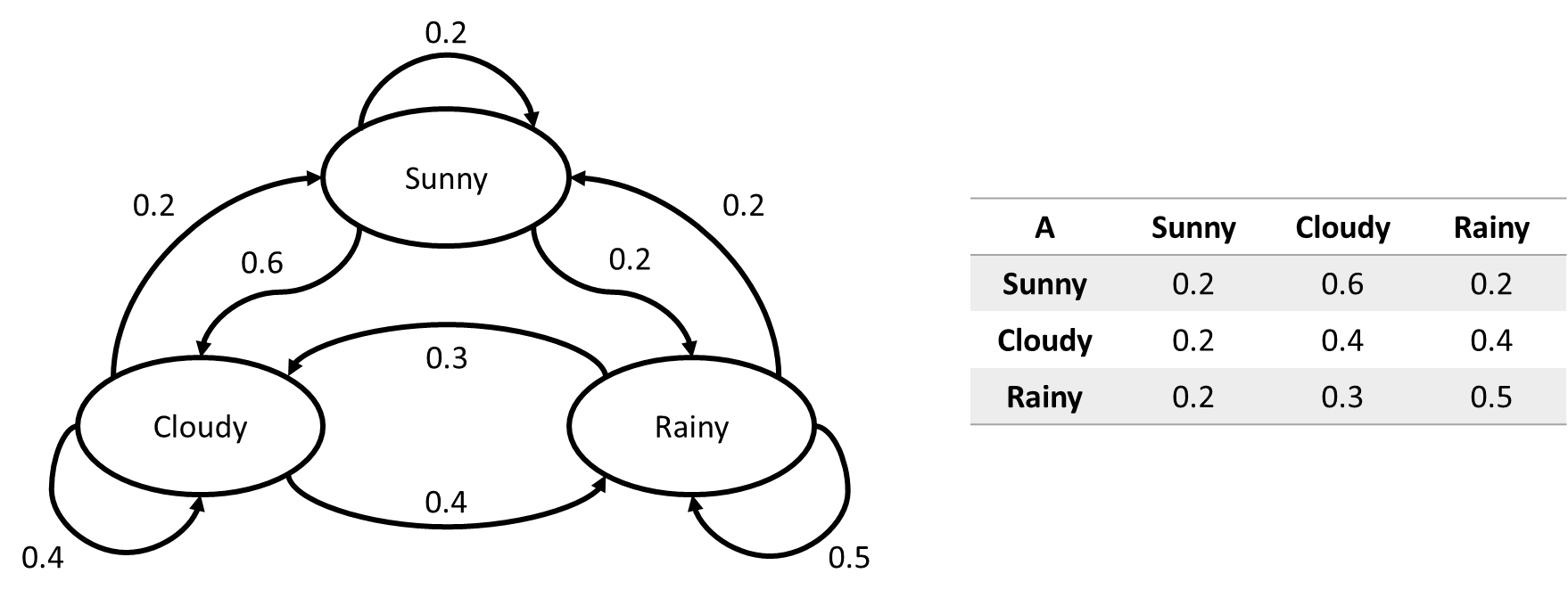


Figure 25. A Markov Model o a simplified weather process with the states and their switch probabilities given on the left and summarised in a state transition matrix A on the right

However, in our case, we do not have a direct observation of the states but rather a sequence of observations generated by a state switching process. This means that each of the states can yield an observation independently [14]. Therefore, we have the situation presented in figure 26 of a hidden state switching process yielding observations. Since we have selected Sunny, Cloudy and Rainy as our observations already let us assume that the weather can switch from Dry to Wet in order to yield the observations.

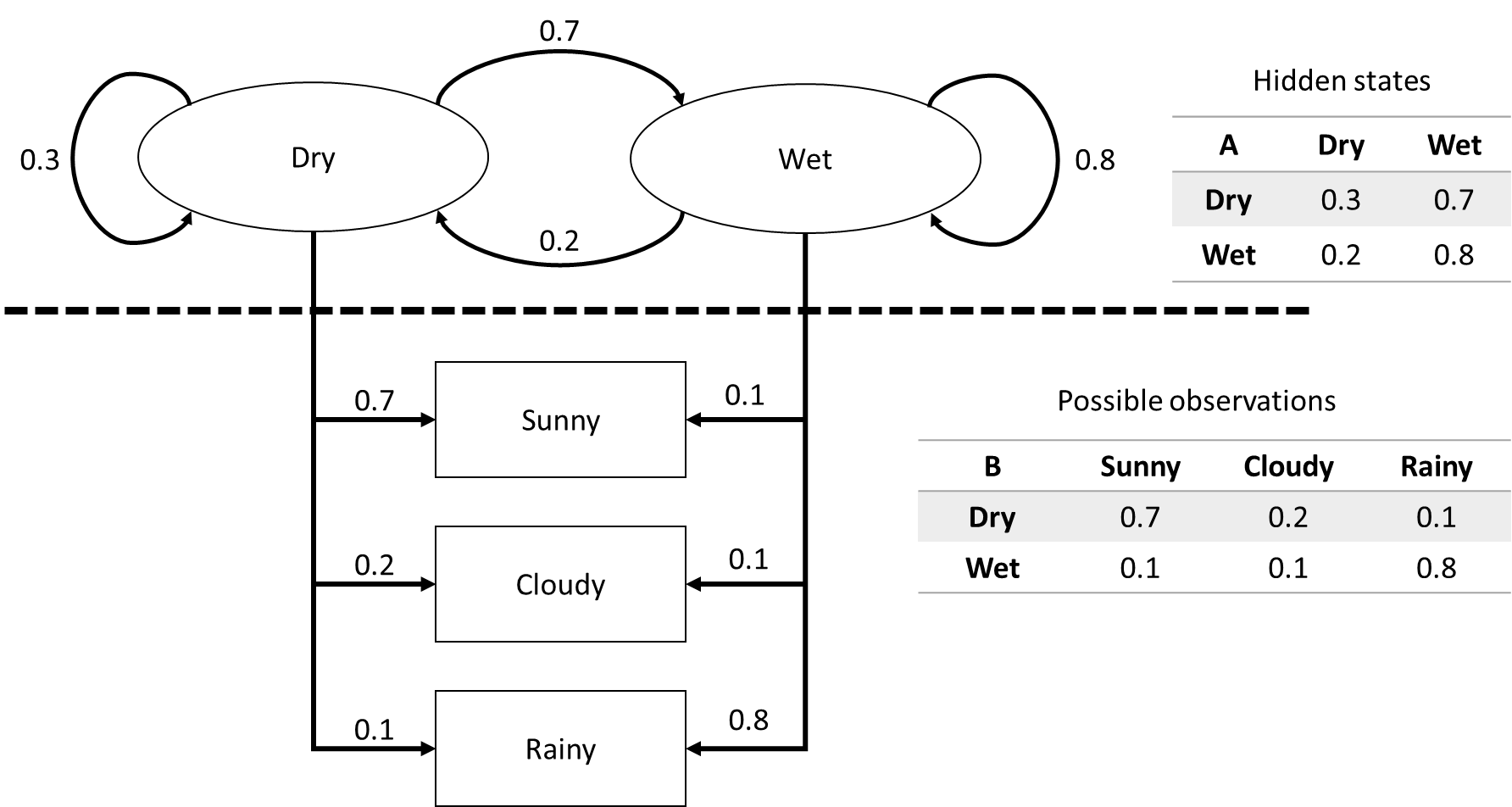


Figure 26. Markov Model with probabilities of switching states given in matrix A and the probabilities of emitting an observation from a given state given in matrix B

As seen from figure 26 an HMM is characterized by its transition matrix A, its emission matrix B and its initial probabilities of being in either state. There are three conventional problems that an HMM can solve, namely evaluation, decoding, and learning. The evaluation problem yields the probability that a given model generated an observed output sequence. The decoding problem yields the most likely sequence of states (the path) the model had to go through to yield the sequence of observations. Finally, the learning problem adjusts the parameters of the model as to have the highest probability of yielding a given sequence of observations [14].

The evaluation problem basically calculates all the possibilities of both state switches under the observation at a given time step and sums them. This becomes excessively computationally expensive with the length of observations and therefore is not very practical. Fortunately, a technique called the Forward Algorithm (FA) where the probabilities of getting to the state at the given time step are saved in a variable called alpha and given by the following three steps [14].

where π is the initial probability of being in either state, b is the value from the emission matrix, X is the observation, a is the value from the state transition matrix, N is the length of the sequence and T is the final time step [14].

The decoding problem is what can ultimately be used for classification in our case. It provides us with the most likely temporal evolution of the states, meaning which was the most likely state at each observation. The Viterbi algorithm provides us with exactly that. It is very similar to the FA, but it keeps an additional variable in which it stores the highest probability for the states at each observation. After the probabilities have been calculated we can backtrack through the values in the additional variable and determine the most likely states. The algorithm is given by the following four steps [14]:

where V is the probability of the path step, W the state at time t-1, and S\* the best state sequence. The max argument selects the highest value that is fed to it and the argmax argument the location of that highest value [14].

Both of these problems are solved if we already have an HMM established. However, in our case, we have no way of estimating the probabilities. Therefore, we need an algorithm which will estimate the model first so we can use the Viterbi algorithm for classification. This is usually done by an expectation-maximization (EM) algorithm to find the maximum likelihood estimate of the parameters. A popular implementation for HMM is the Baum-Welch (BW) algorithm [14].

As with the previous algorithm, the BW is also composed of several steps one of which is part of the FA. However, we have seen that the FA requires an already established HMM. So to start the BW we must either choose random values for the HMM parameters or assign some approximate values if we know something about the nature of the process. In any case, before the algorithm can start the state transition matrix, the emission matrix and the starting probability vector must have some numerical values [14].

It is also easier to think of the observation sequence as a tuple of consecutive values instead of the observation themselves. This means that we include the transition from one observation to the next in the observation sequence as well. In order to create a better estimate of the state transition matrix, we need to calculate two variables. The first one is the sum of all the probabilities of observing the transition of observations under the switch from the first state to the other. This tells us how likely it is that we were in the first state at t=1 and at the second state at t=2 for the given observations but then for all t. This variable is usually given by the Greek letter ksi. The second variable is the highest probability state switch that yielded that observation. This is given by the Greek letter gamma. Then we need to divide the ksi variable by the gamma variable to determine the new value for the state switch between state one and state two. If we do this for all combinations of state switches and then normalize, we would have calculated a new state transition matrix [14].

To calculate a new emission matrix, we simply need to add up the number of times an observation occurred in the state transition variable gamma. Then dividing by the total length of gamma would yield the emission probability for the observation from a given state. Doing this for all observations and normalizing will yield the new emission matrix [14].

Finally, the initial probability vector is simply taken to be the first value of the gamma vector variable. A formal description of the algorithm is given below [14]:

for

Where A is the state transition matrix, B is the emission matrix, π is the initial state probability vector, yt the observation sequence, N the length of the observation sequence, T the last value in a vector, α the FA resulting vector of probabilities, β the BA resulting vector of probabilities.

CHAPTER III

CONCEPTUAL MODEL

After gaining sufficient background on the topic the purpose of the master thesis project can be outlined in more detail. The goals of the research can be set as follows: extracting multiple distinct feature sets from a single channel EEG signal and testing them separately on varying ML algorithms in order to achieve the highest accuracy, sensitivity and specificity; evaluating the single EEG channel which yields the highest accuracy by testing all channels under the selected conditions. Furthermore, the focus of the thesis can be narrowed down even further into: analyzing the influence of the feature sets and evaluating their individual importance, where the importance is given by how high they factor into the decisions made by the algorithm.

Before the goals can be separately described, the limitations of the system must be addressed. The first limitation is concerning the supervised ML algorithms. As already mentioned in the first chapter, even if the performance of the algorithm is perfect (100% accuracy), it will only be as good as the scorer that was used for reference. In order to objectively evaluate the performance of the self-learning algorithm independent data from a different dataset will also be fed for classification. Additionally, a comparison with the non-supervised HMM/Viterbi algorithm will be made.

As described above, understanding the influence of feature sets on the performance of the ML algorithms is the goal. In order to achieve it the project work will follow a closed loop. This loop is defined as follows: extracting a feature set, training a ML algorithm with these features, testing on unknown data, analysing the performance of the system by looking at incorrectly classified data, drawing conclusions on how the feature set might be responsible for the confusion, trying to adjust by extracting features accounting for the confusion.

Features in the time domain are expected to yield a relatively low accuracy. Nevertheless, we expect the occurrence of patterns in the progression of the features correlating to the sleep stages.

Features from the frequency domain are expected to yield an overall better accuracy since they are directly related to how a human scores the data in the PSD case. However, we expect to see a poor distinction between Stage 1 and REM sleep since their characteristics are very similar. Again a pattern in the data is to be expected in correlation to sleep stages. It will be interesting to examine the correlation between the AR parameters and sleep stages as the exact characteristic which they represent is somewhat obscure.

Features from the time-frequency domain are expected to yield the best results from all techniques. Again a clear correlation between the values of the features and the sleep stages should be observed. In particular, the CWT features should provide the fullest description of the epochs.

The non-linear features might prove too time-consuming for extraction. Furthermore, they are a measure which presents information similar to the time domain features. It is unknown if the benefit of having these features will outweigh the cost of extracting them.

The development of the HMM is ultimately aimed at examining whether the sleeping brain follows a process described by a Markov process. However, answering that question is quite possibly involving many more parameters and experiments. As far as this study is concerned, the HMM will be used as a comparison for the evaluation of the ML algorithm outside of the limits imposed by the reference scorer. In a way, this can be interpreted as simulating the scores given by an independent scorer on the same data.

It is to be expected that the difference in accuracy, sensitivity, and specificity will vary between different sets of features. Other studies suggest that the best performance will be achieved by the RF algorithm. Moreover, we expect that there will be difference in the performance parameters in between the EEG channels. Additionally, the effect of dimensionality reduction techniques such as principal component analysis (PCA) is to be examined. Finally, computational performance techniques such as parallel processing might be discussed.

CHAPTER IV

RESEARCH DESIGN

4.1 Criteria, data, and equipment

4.1.1 Criteria and equipment

The performance will be evaluated by comparing the Accuracy, Specificity, and Sensitivity of each technique, given by the following formulas:

where:

TP: True Positives (data which has been correctly identified)

FP: False Positives (data which has been incorrectly identified)

FN: False Negatives (data which has been incorrectly rejected)

TN: True Negatives (data which has been correctly rejected)

The ML techniques will be coded in a Python 3.7 environment running on an Intel® Core™ i7-4710MQ CPU @ 2.50GHz with 8.00 GB of RAM memory with a 64-bit Operating System.

4.1.2 Data

Hut lab Dataset

The Hut lab dataset was provided by the HUT sleep lab at the Chronobiology department at the Rijksuniversiteit Groningen. The data has been collected over 2 years (from November 2014 until May 2016). The number of subjects varies on the length of the epochs. For the 10 second epochs, the number of subjects is 50. For the 30-second length epochs, the number of subjects is 40. The data is sampled at 128 Hz. The data is collected for full night sleep and the length varies between 10 to 14 hours. The EEG electrodes from which the data is collected are the Cz, Fpz, Oz, C3, and C4.

DREAMS dataset

The data from the DREAMS sleep EEG subjects dataset contains the full-night sleep EEG data of 20 subjects. The data is sampled at 200 Hz and the epochs are of 5-second length. The length of the signals ranges from 8 to 10 hours. The electrodes which are used to record the EEG signals are Fp1, Cz, O1, Fp2, O2, and Cz2.

4.2 Experiments

4.2.1 Feature sets evaluation over different channels

The first set of experiments is the evaluation of different feature set performance. The procedure that was followed for feature extraction in all domains for all datasets was the following.

1. Convert the. edf files and the hypnogram .txt files containing the EEG signals into .csv files.
2. Calculate the length in data points of each sample based on the sampling frequency and save them as the columns of a new .csv file.
3. Use a python library implementation to calculate the features based on the techniques described in section 2.2.2
4. Save the features for each epoch as rows in a new .csv file and append the score given by the hypnogram files.
5. Remove any rows (epochs) that have invalid values (NULL, inf, or epoch scores higher than 6)
6. Combine the feature files for all subjects into 1 large file containing all epochs for all subjects.
7. Separate the data into training and testing sets with a ratio of 70:30
8. Feed the training data into an ML algorithm implementation (RF taken as a base reference) to train a model.
9. Predict the scores of the testing data based on the trained model.
10. Calculate the Accuracy, Sensitivity, and Specificity of the model.
11. Repeat for all channels of the data

This procedure was performed for two separations of the data. The variable was the number of scores as the first time the number of scores remained as in the original data (7 classes) and for the other case the number of classes was reduced to 3 in which case the movement and wake were combined into one class with the NREM and REM forming the other 2.

4.2.2 Evaluation of different ML algorithms

In this experiment the goal was to examine which of the ML algorithms gives us the best results. In order to do that the same procedure as in the previous experiment was followed with the difference being the type of model being trained.

4.2.3 Individual feature evaluation

After the feature sets have been evaluated as a whole and the best ML algorithm selected, a deeper look at the individual features themselves was done. The files with different features were combined into one in order to have a full feature set form all domains. The importance of the features was examined by looking at their weights for the SVM, the number of decisions that stemmed from a particular branch for the RF, and by evaluating the distance for the KNN. In every case the implementations for the algorithms from skit-learn python library were used. After the evaluation was done, Principal Component Analysis was performed to create a feature space with the highest variation. Then the ML algorithms were trained again to examine if any improvement on the accuracy was achieved.

4.2.4 Hidden Markov Model evaluation

To be performed…

4.3 Hypotheses

4.3.1 Time domain feature hypothesis:

Since we said that the accuracy yielded by the time features is not to be expected in the higher end it is reasonable to concentrate our hypotheses for this section on how the time domain features will correlate with the sleep stages. However, the hypothesis that each of the statistical moments will represent the signal better than the raw data remains.

The mean is perhaps not the most useful feature, because in most case its value in most cases will be around due to the fact that sleep EEG is often described as a stochastic process. That being said, the signals we have at our disposal are actual recordings from sleep experiments which include artifacts. It has been argued that these artifacts are created by imperfections in the recording due to movement of the electrode, sweat or another source of noise. They often take the form of a highly positive or negative peak which does not necessarily have a mirrored image later on to equalize the mean. Therefore, we hope that the mean will be useful when segregating the artifacts and consequently the movement stages. It must be mentioned that we do not explicitly support the conclusion that artifacts are mostly due to movement but nevertheless the data we have was scored under that assumption.

The standard deviation explains the spread of the data. We believe that this is a suitable feature because different sleep stages are characterized by waves of varying amplitude. SWS is typically represented by a lower frequency but higher amplitude waves, while stage 1 and REM sleep is characterized by higher frequency but lower amplitude waves. Thus, the standard deviation value should help distinguish between those.

The third moment, skewness is a measure which, as the name suggests, shows how skewed the distribution of values is. This essentially tells us is how the data is divided between the mean, whether there are values which are extremely abundant on one side even though the rest is standardly distributed. Taking into account that our data oscillates around zero, the skewness factor tells us how much time the signal spent above and below the zero time and more specifically if there was something that caused the data to be over or under the zero-line for a significant portion of the epoch. In this sense, skewness is directly related to the mean of the population and could be representative of the artifacts in the signal.

The kurtosis feature shows how much of the data is close to the center of the distribution and how much of it is further away. As with the relationship between mean and skewness, the kurtosis feature and the standard deviation of the population share a relation. Again it shows the distinction between the high-amplitude and low-amplitude waves, meaning distinguishing between SWS and faster oscillating waves.

We hope that the zero-crossing rate will help distinguish between the different sleep stages since Rechtshafen and Kales heavily rely on the cps (crossing per second) factor to define them [5]. It is, however, important to distinguish between zero-crossing rate and frequency since they are not the same concept.

Hjorth activity gives us information concerning the amplitude of the signal. This should mean that high values of activity correspond to SWS and low levels to REM sleep. In Hjorth, mobility higher values should be related to SWS while lower values are connected to REM. In Hjorth complexity, higher values should be associated with SWS and lower with REM. As we can see we expect a clear pattern of higher Hjorth parameters for the SWS and lower for the REM stages.

4.3.2 Frequency features

The features extracted from the frequency domain are expected to yield a higher accuracy than the ones from the time domain. The basis for that assumption comes from the fact that the frequency bands which are part of the features are the characteristics which human scorers use. It is obvious that the expectation of high power in the Delta band will be connected to SWS and rises in the Theta and Alpha bands connected to S1, S2, and REM.

The parametric features are also expected to yield a better performance than the time domain ones but the performance relative to the non-parametric is difficult to predict. In any case, the expectation is that the SWS will have lowered values for the higher coefficients.

4.3.3 Time-frequency features

The features based on the time-frequency domain are expected to have the highest accuracy of all. In this case again as for both the STFT and the CWT, the expectation is that the power in the frequency bands will follow the logic set by the rules for scoring, namely high power in the Delta band for SWS and high power in the Theta and Alpha bands for S1, S2, and SWS.

4.3.4 ML algorithm performance

Literature suggests that the performance of the Random Forest classifier will be the best. Nevertheless, k-nearest neighbors and support vector machines have also shown promise in their respective accuracies. However, it is impossible to make predictions for the separate algorithms without reviewing the characteristics of the data first.

CHAPTER V

RESEARCH RESULTS

5.1 Feature set evaluation and channel evaluation

5.1.1 Time domain features

As we have already discussed the features extracted from the time domain are:

* Mean
* Standard deviation
* Minimum value
* Maximum value
* Zero-crossing rate
* Hjorth mobility
* Hjorth complexity

Figure 5.1 illustrates the average value of each of these features for the different classes for our classifier algorithm. This gives an initial idea of how well these features would segregate the classes. We can see that the means of the Zero-crossing rate, minimum value, and maximum value have the highest variance suggesting that they will have the highest importance when training a model.

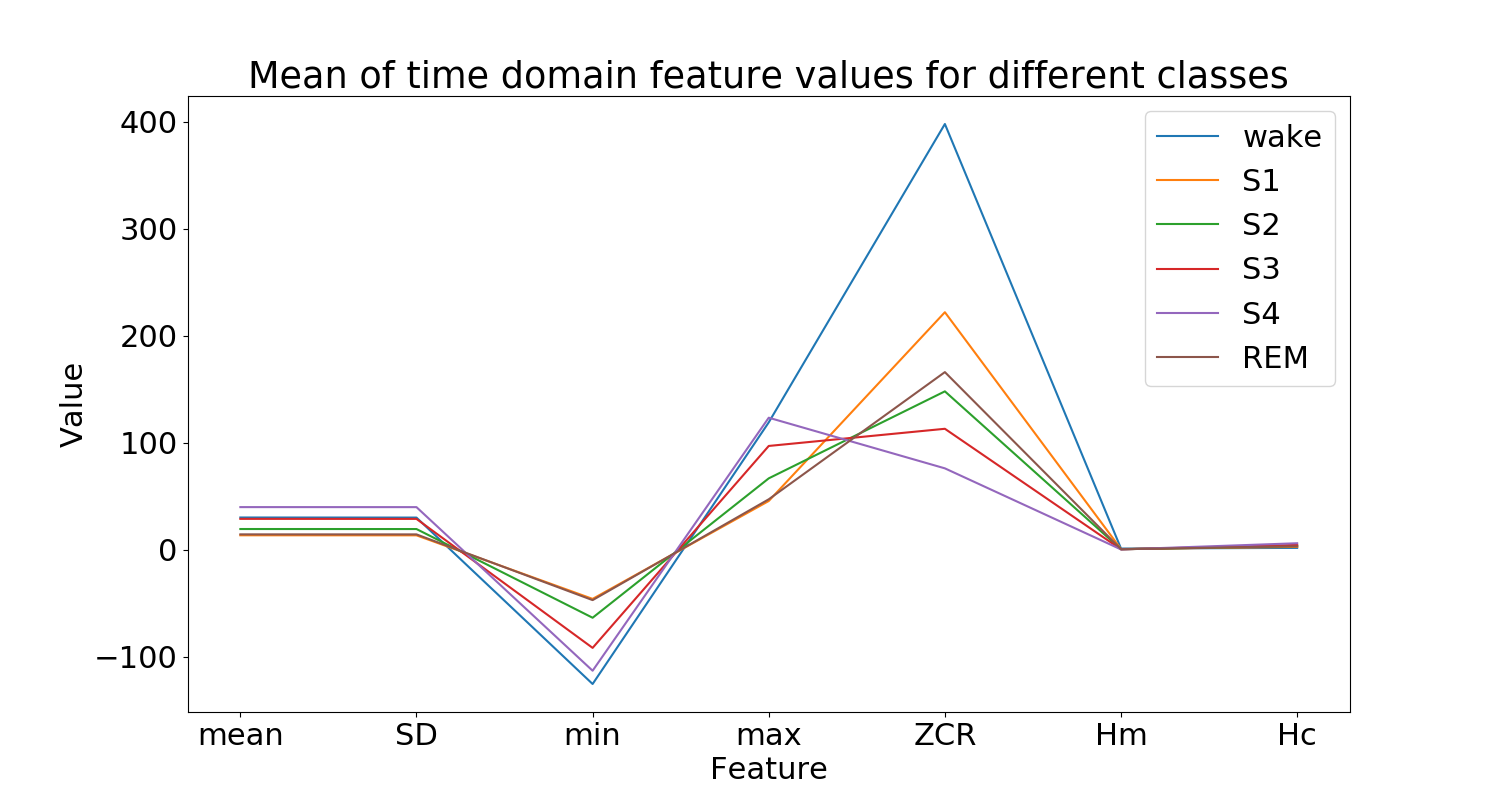


Figure 27. Feature variation across different classes. The features with highest variance in mean values are best suited for segregation of the classes.

While looking at the mean is a useful first step, it can be deceiving because the y-axis has one range for all features. It is perhaps better to look at the variance in distributions. Therefore, the pair plot shown in figure 5.2 is a better representation of how well the feature sets will segregate classes. As with figure 5.1 here we can also see that the most variance is found in the pairs containing the ZCR, minimum, and maximum. However, here Hjorth mobility pairs also have a larger distribution.

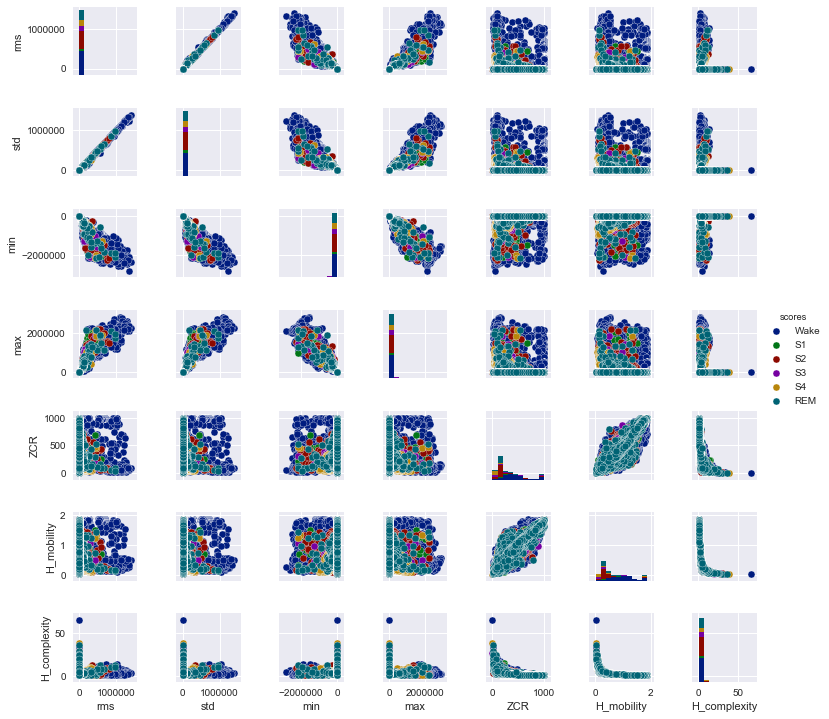


Figure 28. Pair plot of the time domain features. The plot shows the relationship between separate feature sets. Ideally the data points with the same colours would form separate clusters, therefore larger variance is desirable.

Now that we have a better idea of the characteristics of the feature set, a model can be trained to classify the sleep stages. In this case a RF and an KNN model has been trained to classify the sleep stages. A separate model is trained for each channel. The ratio of training to testing data is 80:20. For the KNN a k of 30 is chosen. A 10-fold cross-validation is performed in order to achieve an unbiased result for accuracy. We see from the results in figure 5.3 that the performance of the RF algorithm is greater than the KNN for all channels in both a 3-class and a 6-class classifier. Additionally, we see that the best performance is given by the RF for 3 classes on the Oz-C3 channel with a value of 80.8 ± 0.5. It is worth noting that there is a difference in the channel giving the highest accuracy between the RF and the KNN. While the Oz-C3 gives the greatest accuracy for the RF algorithm, the KNN works best under the features from the Oz-Fpz. However, both algorithms yield the lowest accuracy under the Cz-C4. The Cz-C3 channel yields similar accuracy. These two can be easily distinguished from the rest as they make for a much lower accuracy than the rest. However, it is also worth noting that the C3-C4 channel shares their level of accuracy only at the 3 classes RF model. Looking at the system for electrode positioning we can clearly see that all three electrodes (Cz, C3, and C4) are on top of the head, as their abbreviations suggest: C for Centre.

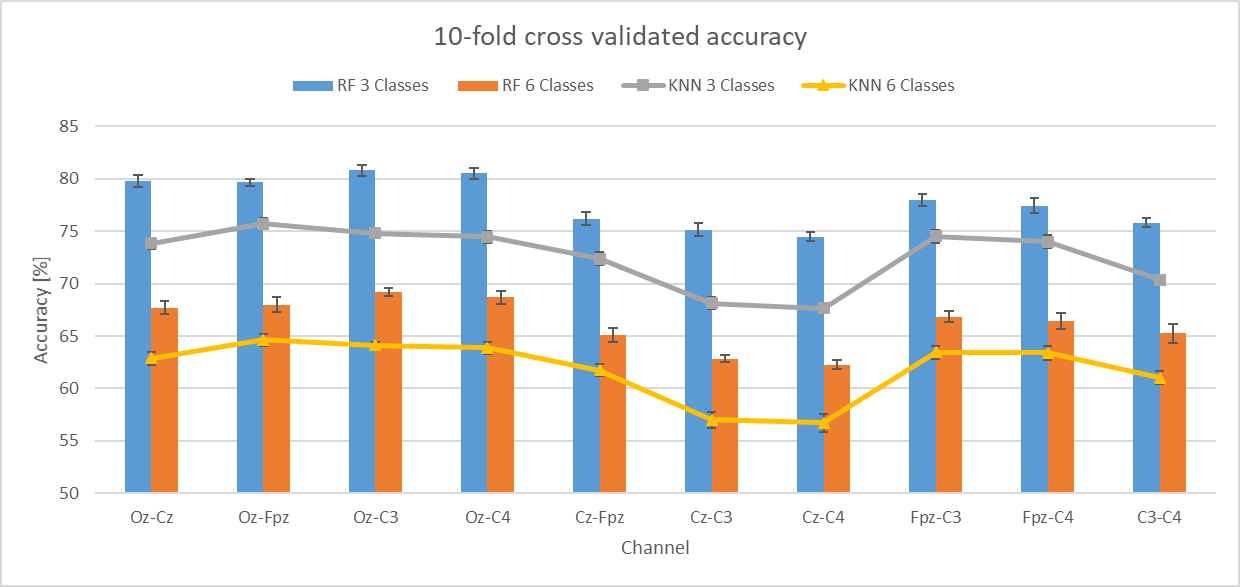


Figure 29. 10-fold cross-validated Accuracy with error ranges for each EEG channel of the 10 second Hut lab data for the time domain feature set under both a Random Forest and a K-nearest neighbour (k=30) classifiers for both 3 and 6 classes.

While this is an impressive result, given the feature set we fed into the algorithm it is important to understand where the errors come from. The confusion matrices shown in figure 5.4 helps analyse this. We can see from the first matrix that the Wake class has a very high rate of true positives. The S2 and S4 calsses are also relaively well predicted with rates of 0.7 in the 6 class classifier. The other 2 NREM classes are more often misclassified than correctly predicted. As for the REM class, it is onlyy classified correctly roughly half of the time. These observations lead to the conclusion that combining the NREM stages into one would account for some of the errors. As we can see formt he matrix on the right and the accuracy from figure 5.3, this is indeed the case. In this case the classification of Wake and NREM is above 90% correct, but the issue for the REM class continues. One solution to this problem is to extract features that better characterize the REM stages such that they are more easily segregated.

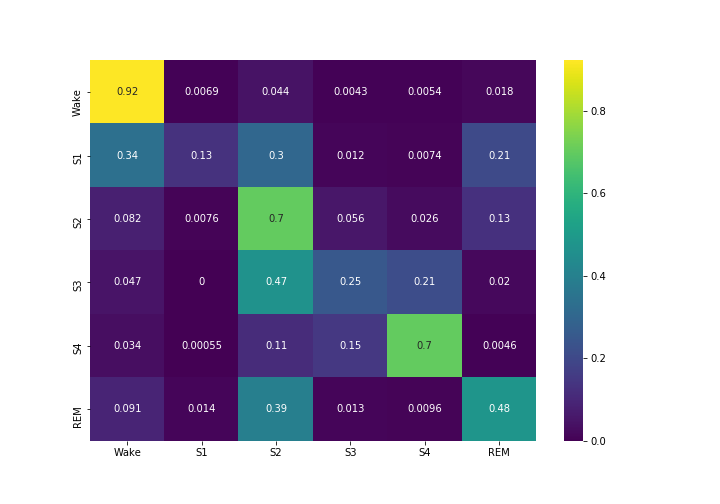
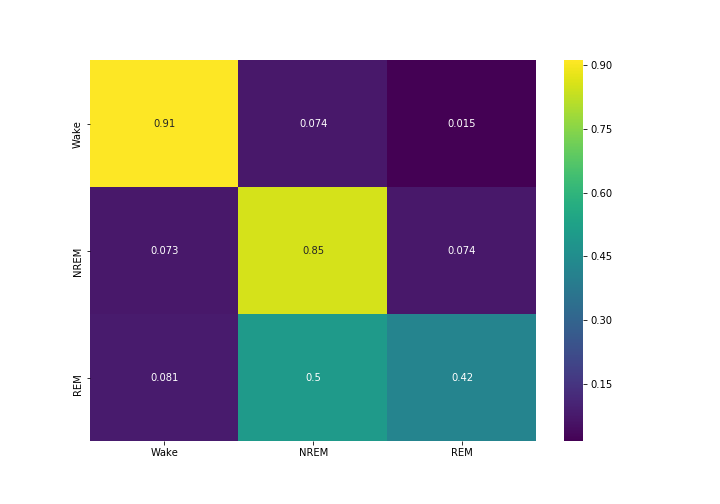


Figure 30. Normalized confusion matrix for 6 class (left) and 3 class (right) RF classifiers of the Oz-C3 channel for the time-domain 10 second Hut lab data model.

Before moving onto the next type of features a final look at what characterizes the performance of the algorithms in the form of the Sensitivity and Specificity of the model can be useful. Table 5.1 summarizes these characteristics for the best and the worst models and their respective channels. The full table can be seen in Appendix A. The table shows the best results, shown in green, and their respective changes in the worst results shown in red. Table 5.1 confirms the observations from the confusion matrices that stages Wake, S2, and S4 have are recognized better, while the rest suffer from poor sensitivity. We can see that logically, there is a drop in sensitivity across almost all classes for the worst case. Specificity however, is barely affected at all, with only a few values slightly dropping. Both of these observations, show us that the models are too specific. This means that almost all of our negatives are correctly labelled as negatives. However, it also means that many positives are also labelled as negatives.

Table 2. Sensitivity and Specificity of the best (Oz-C3 for RF and Oz-Fpz for KNN) channels and the worst channel. The best performance data is shown in green while the corresponding changes for the worst channel are outlined in red.



Looking at the data can give us an idea of why there is a high specificity. Typically, the RF and KNN implementations in Python have a classification threshold of 0.5 by default. What this means is that the probability of something being put in a class is 50%. This works very well in balanced binary classification problems. However, in our case, the data is largely unbalanced as seen from the number of epochs in each sleep stage. Additionally, the problem is not binary which is unfortunate, because adjusting the decision threshold is a solution in unbalanced binary problems. In this case however, even if the problem is made binary by using a one vs all method where each class is taken versus a combination of all the rest, multiple optima can be found for a threshold value. This is confirmed by the ROC plot shown in figure 5.5. Logically, the area under the curve for the Wake (class 0) and S4 (class 4) is highest as also given by their respective sensitivities. Moreover, sleep stage 1 has the lowest area, while the S2 and S3 share similar results. It is also seen that the REM curve is also in the same range as S2 and S3. This means that the threshold value affects the S1 class the most. This means that multiple legitimate S1 epochs were misclassified which can also be seen from the S1 row in the confusion matrix. This confirms the earlier hypothesis that features describing the S1, S2, S3 and REM better are required for a better model. Additionally, changing the threshold value in the algorithm implementation yields the same results as the original one.

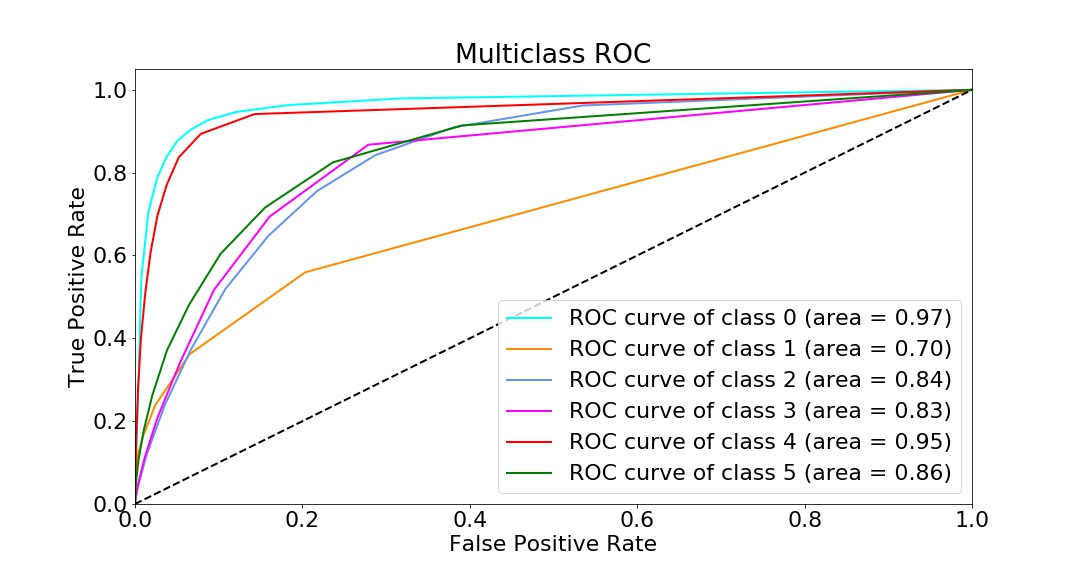


Figure 31. ROC curves for all classes under the Oz-C3 channel for 6 classes using RF. The classes go as follows: 0-Wake ;1-S1;2-S2; 3-S3; 4-S4; 5-REM

At the end of the time domain feature analysis it is important to show which features have the highest influence on the decisions made by the algorithm. The RF algorithm can also yield the rate of classification decisions originating from each feature, meaning how many times the feature was responsible for classification. There are 10 separate models (1 per EEG channel) which assign different weights to each feature. In order to rate the importance overall a system which assigns scores to the feature was made. The most important feature receives a score of 7 and the least important a score of one. Summing the scores for each channel results in the values presented in figure 5.6. We see that the ZCR, Hjorth mobility and Hjorth complexity have a higher overall importance than the rest.

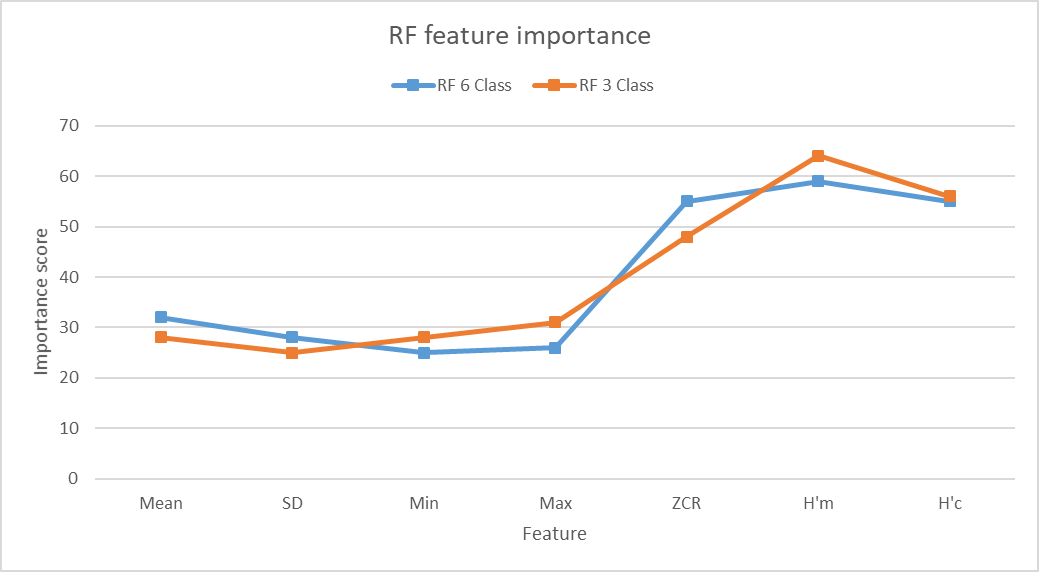


Figure 32. Feature importance for the RF algorithm derived through summing the importance of the features across all channels.

5.1.2 Parametric features

As with the features from the time domain we first look at the distribution of the features.

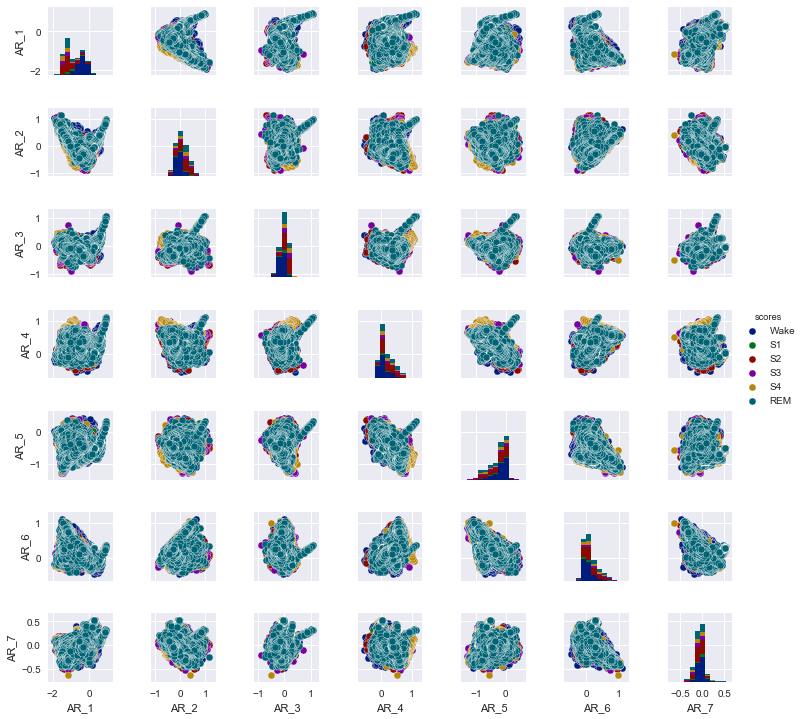
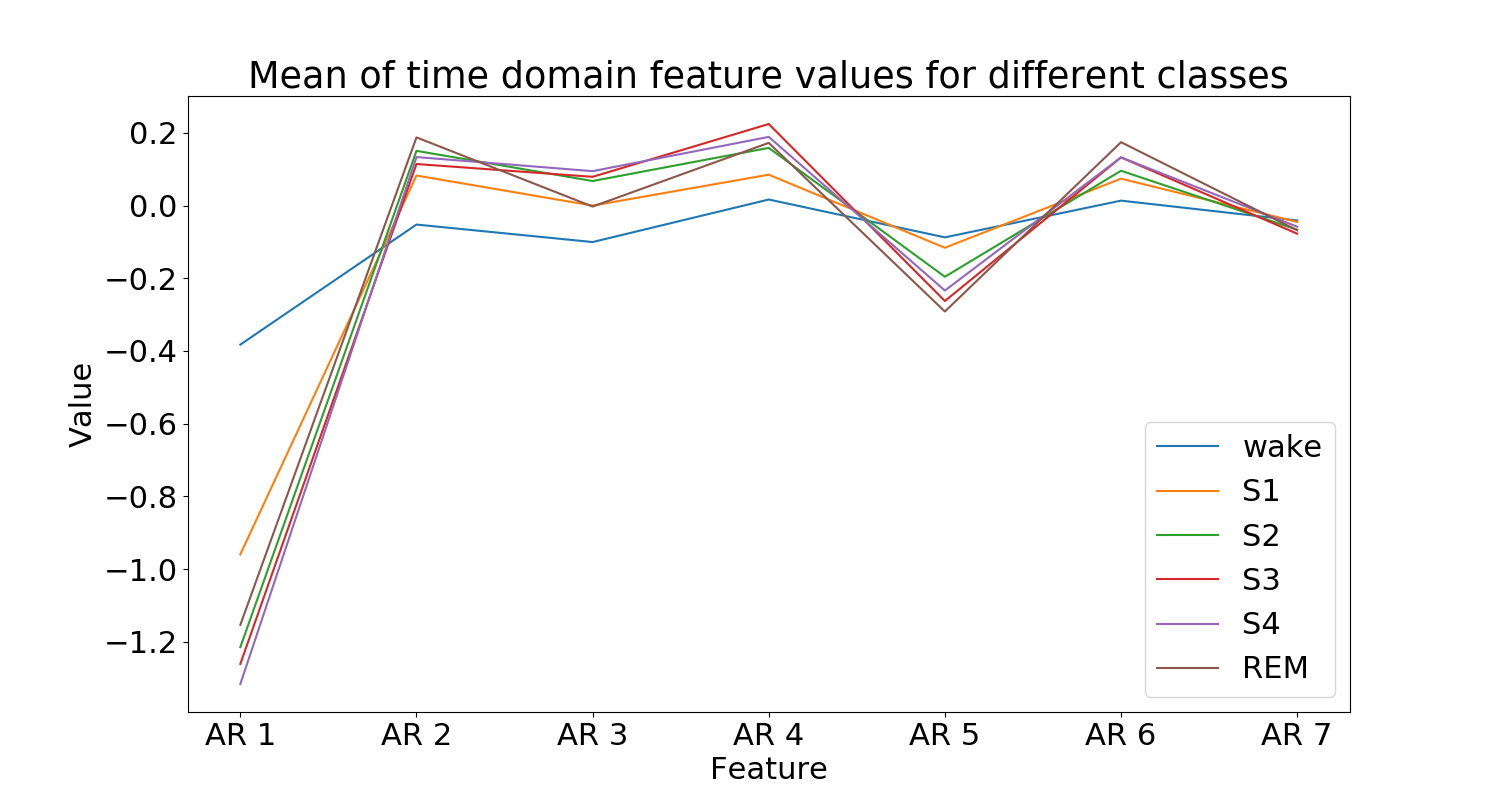


Figure 33. Means of the features for the different classes (top), and the respective pair plot showing the distribution of the features in respect to each other.

These distributions suggest that the first defining weight will have the highest importance because of the largest distribution that it presents.

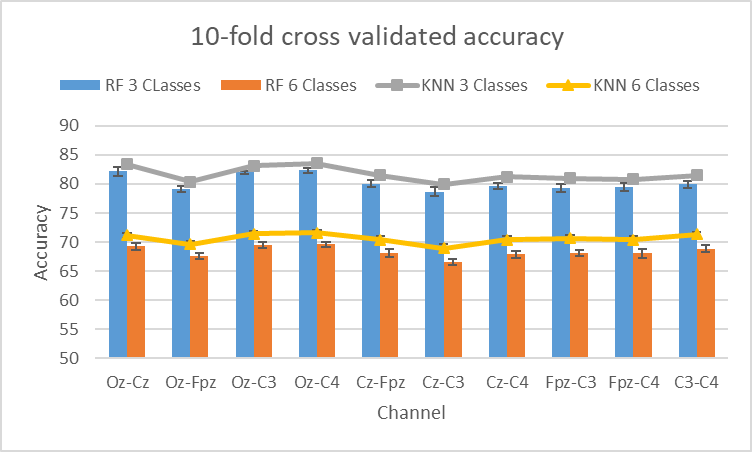


Figure 34. 10-fold cross-validated Accuracy with error ranges for each EEG channel of the 10 second Hut lab data for the parametric frequency domain feature set under both a Random Forest and a K-nearest neighbour (k=30) classifiers for both 3 and 6 classes.

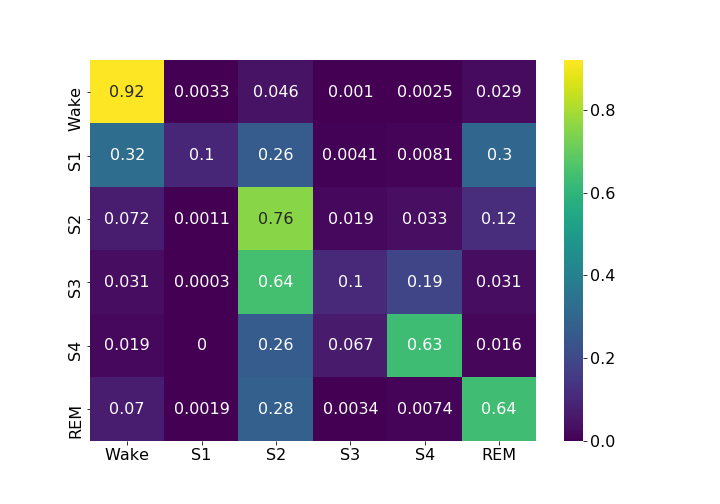


Figure 35. Normalized confusion matrix for 6 class (left) and 3 class (right) KNN classifiers of the Oz-C4 channel for the parametric frequency domain 10 second Hut lab data model.

Table 3Sensitivity and Specificity of the best (Oz-C4) channels and the worst (Cz-C3) channel. The best performance data is shown in green while the corresponding changes for the worst channel are outlined in red.



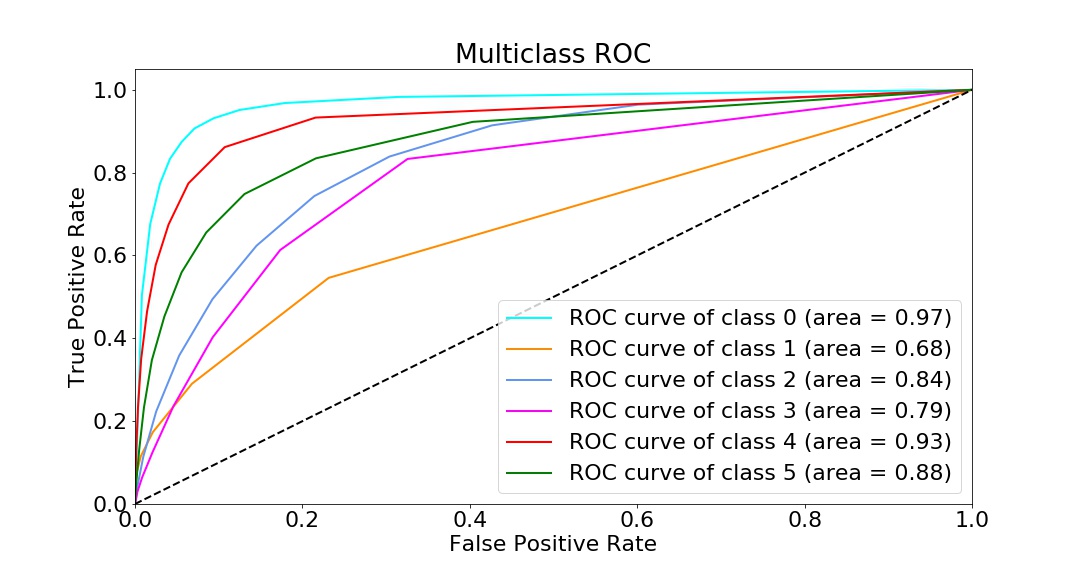


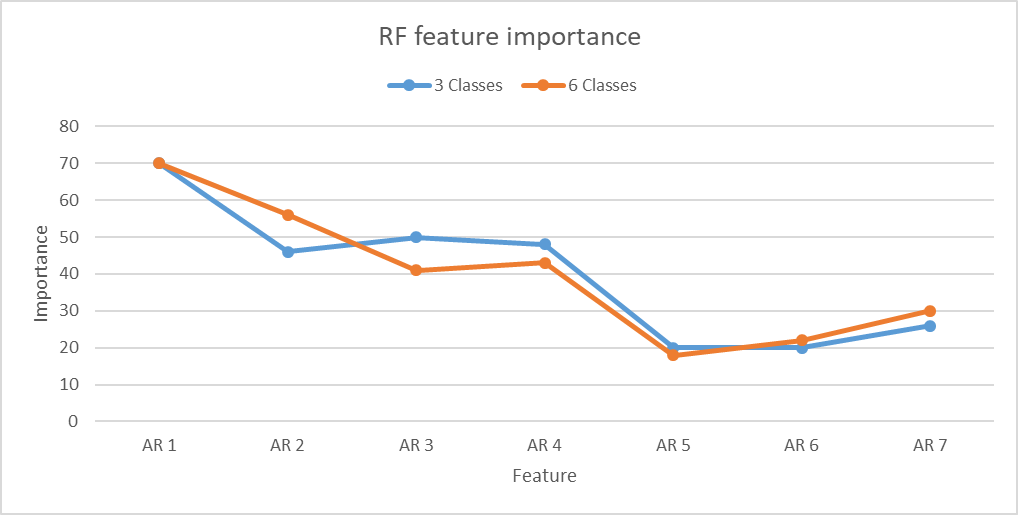
Figure 36. ROC curves for all classes under the Oz-C3 channel for 6 classes using RF. The classes go as follows: 0-Wake ;1-S1;2-S2; 3-S3; 4-S4; 5-REM 

Figure 37. Feature importance for the RF algorithm derived through summing the importance of the features across all channels.

5.1.3 Continuous wavelet transform features

The procedure of describing the time-frequency domain features will be slightly different. Since there are many more features than in the previous section the pair plots will be given in multiple figures as a function of frequency bands.

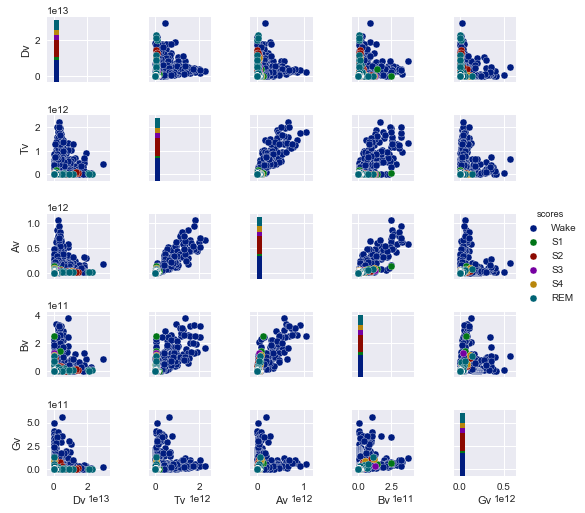
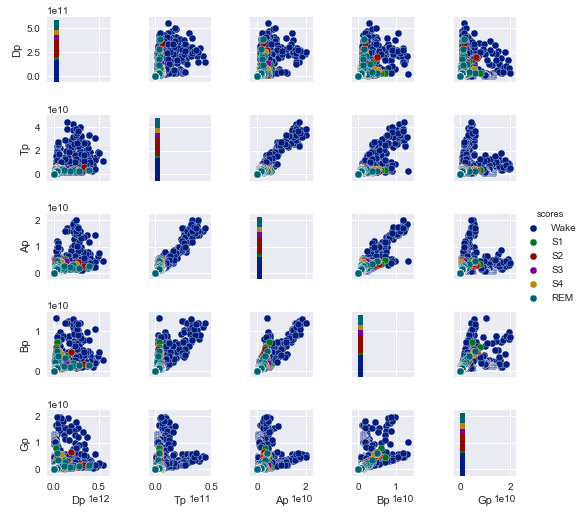
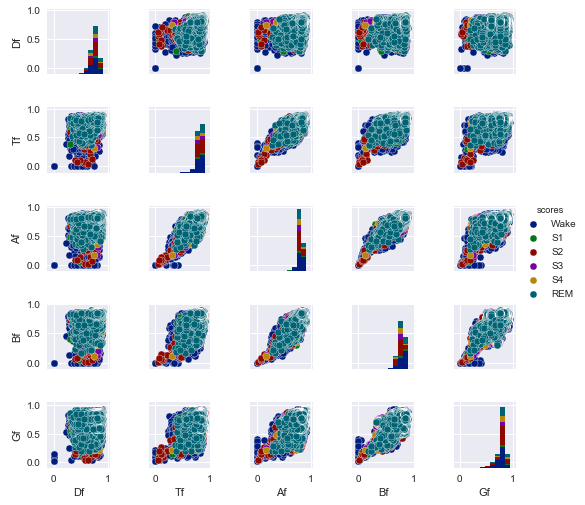
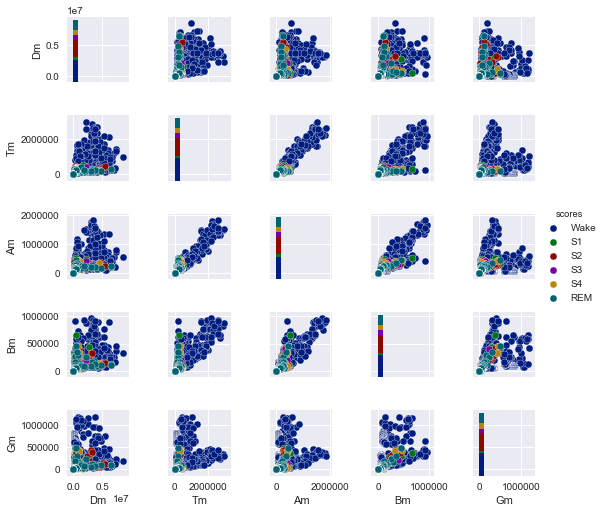


Figure 38. distributions of different features across the bands: means (top left), middle-crossing rate (top right), total band power (middle left), inequality (middle right), variance (bottom left), spectral edge frequency (bottom right)

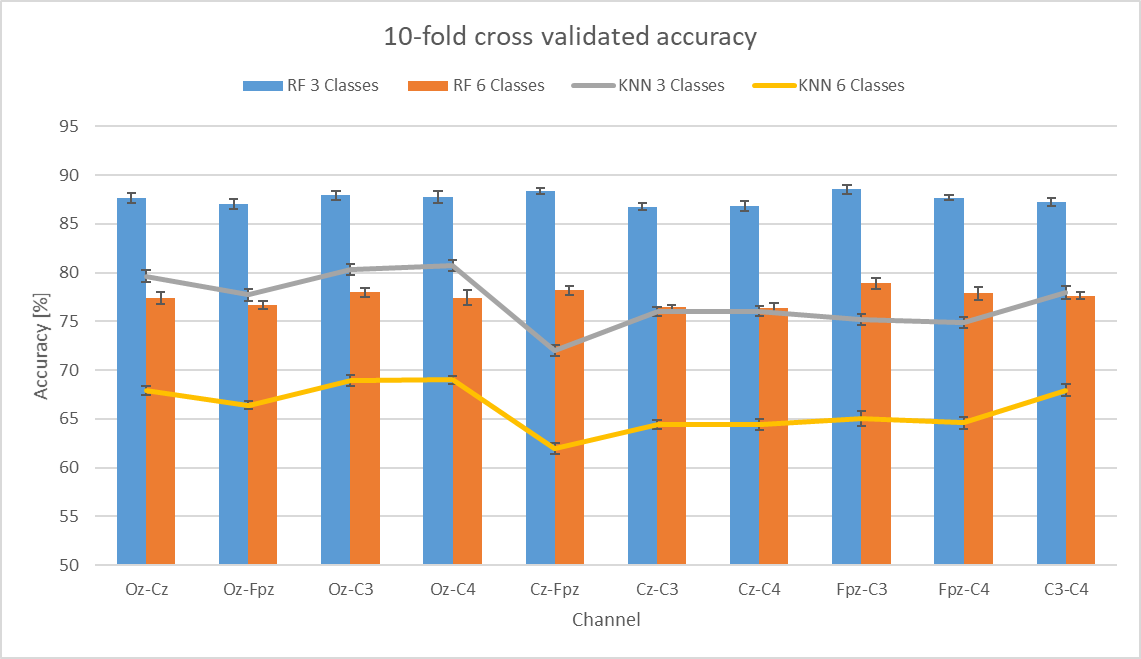


Figure 39. 10-fold cross-validated Accuracy with error ranges for each EEG channel of the 10 second Hut lab data for the CWT time- frequency domain feature set under both a Random Forest and a K-nearest neighbour (k=30) classifiers for both 3 and 6 classes

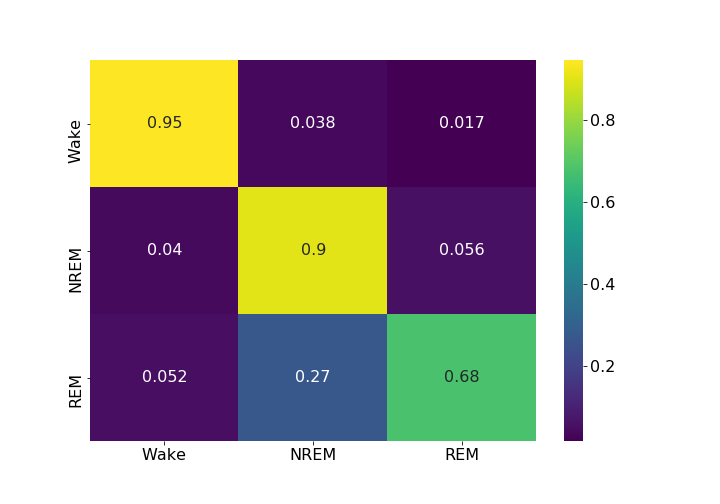
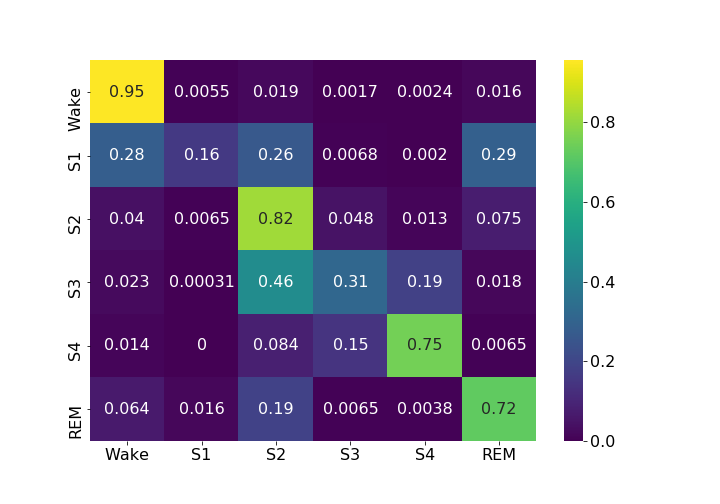


Figure 40. . Normalized confusion matrix for 6 class (left) and 3 class (right) KNN classifiers of the Fpz-C3 channel for the parametric frequency domain 10 second Hut lab data model.

Table 4. Sensitivity and Specificity of the best (Fpz-C3 for RF and Oz-C4 for KNN) channels and the worst (Cz-C4 for RF and Cz-Fpz fro KNN) channel. The best performance data is shown in green while the corresponding changes for the worst channel are outlined in red



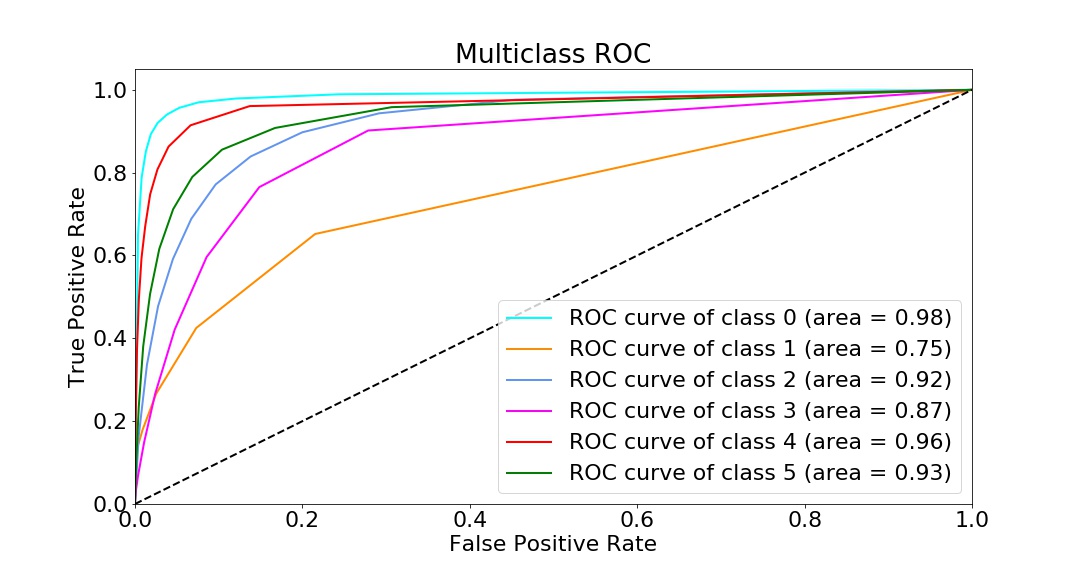


Figure 41. ROC curves for all classes under the Oz-C3 channel for 6 classes using RF. The classes go as follows: 0-Wake ;1-S1;2-S2; 3-S3; 4-S4; 5-REM

Do feature evaluation of some sort

5.2 Optimization of performance

5.2.1 Parallel processing

5.2.2 ML Algorithm optimization

5.3 ML performance evaluation and comparison

CHAPTER VI

CONCLUSIONS & RECMMENDATIONS

Provide appropriate conclusions after carefully analyzing the data. Still to be decided…

LIST OF DEFINITIONS AND ABBREVIATIONS

polysomnography PSG

electroencephalography EEG

electrooculogram EOG

electromyogram EMG

machine learning ML

Rechtschaffen and Kales R&K

American Academy of Sleep Medicine AASM

Rapid Eye Movement REM

Non-Rapid Eye Movement NREM

Suprachiasmatic Nucleus SCN

Support Vector Machine SVM

K-nearest neighbor KNN

Random Forest RF

Decision Trees DT

Hidden Markov Models HMM

Forward Algorithm FA

Backward algorithm BA

Baum-Welch algorithm BW

Expectation-maximization EM

True positives TP

True negatives TN

False positives FP

False negatives FN

REFERENCES

[1] S. Paruthi *et al.*, “Recommended amount of sleep for pediatric populations: A consensus statement of the American Academy of Sleep Medicine,” *J. Clin. Sleep Med.*, 2016.

[2] S. . D. Banks D. F., “Behavioral and physiological consequences of sleep restriction,” *Behav. Physiol. consequences sleep Restrict.*, 2007.

[3] S. A. Keenan, “An Overview of Polysomnography,” in *Review of Sleep Medicine*, 2007.

[4] Q. S. for the A. A. Iber C, Ancoli-Israel S, Chesson AL Jr. *et al.*, “The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications,” in *AASM Manual for Scoring Sleep*, 2007, pp. 1–59.

[5] A. Rechtschaffen and A. Kales, “A manual of standardized techniques and scoring system for sleep stages of human subjects,” *Washington, D.C. U.S. Gov. Print. Off.*, vol. NIH Public, p. 12, 1968.

[6] R. S. Rosenberg and S. Van Hout, “The American Academy of Sleep Medicine inter-scorer reliability program: Respiratory events,” *J. Clin. Sleep Med.*, vol. 10, no. 4, pp. 447–454, 2014.

[7] K. A. I. Aboalayon, M. Faezipour, W. S. Almuhammadi, and S. Moslehpour, “Sleep stage classification using EEG signal analysis: A comprehensive survey and new investigation,” *Entropy*, vol. 18, no. 9, pp. 50–70, 2016.

[8] R. Boostani, F. Karimzadeh, and M. Nami, “A comparative review on sleep stage classification methods in patients and healthy individuals,” *Comput. Methods Programs Biomed.*, vol. 140, pp. 77–91, 2017.

[9] K. A. I. Aboalayon, W. S. Almuhammadi, and M. Faezipour, “A comparison of different machine learning algorithms using single channel EEG signal for classifying human sleep stages,” *2015 Long Isl. Syst. Appl. Technol.*, pp. 1–6, 2015.

[10] B. Koley and D. Dey, “An ensemble system for automatic sleep stage classification using single channel EEG signal,” *Comput. Biol. Med.*, vol. 42, no. 12, pp. 1186–1195, 2012.

[11] A. R. Hassan and M. I. H. Bhuiyan, “Automatic sleep stage classification,” *2nd Int. Conf. Electr. Inf. Commun. Technol. EICT 2015*, pp. 211–216, 2016.

[12] K. Šušmáková and A. Krakovská, “Discrimination ability of individual measures used in sleep stages classification,” *Artif. Intell. Med.*, vol. 44, no. 3, pp. 261–277, 2008.

[13] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. 1995.

[14] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[15] N. Goel, H. Rao, J. S. Durmer, and D. F. Dinges, “Neurocognitive consequences of sleep deprivation,” *Seminars in Neurology*, vol. 29, no. 4. pp. 320–339, 2009.

[16] W. D. S. Killgore and M. Weber, “Sleep deprivation and cognitive performance,” in *Sleep Deprivation and Disease: Effects on the Body, Brain and Behavior*, 2014.

[17] J. Lim and D. F. Dinges, “Sleep deprivation and vigilant attention,” in *Annals of the New York Academy of Sciences*, 2008, vol. 1129, pp. 305–322.

[18] D. F. Dinges and N. B. Kribbs, “Performing while sleepy: effects of experimentally-induced sleepiness,” in *Sleep, Sleepiness and Performance*, 1991.

[19] P. Philip, J. Taillard, C. Guilleminault, M. A. Quera Salva, B. Bioulac, and M. Ohayon, “Long distance driving and self-induced sleep deprivation among automobile drivers,” *Sleep*, 1999.

[20] S. Taheri, L. Lin, D. Austin, T. Young, and E. Mignot, “Short sleep duration is associated with reduced leptin, elevated ghrelin, and increased body mass index,” *PLoS Med.*, 2004.

[21] K. Spiegel, R. Leproult, and E. Van Cauter, “Impact of sleep debt on metabolic and endocrine function,” *Lancet*, vol. 354, no. 9188, pp. 1435–1439, 1999.

[22] O. Tochikubo, A. Ikeda, E. Miyajima, and M. Ishii, “Effects of insufficient sleep on blood pressure monitored by a new multibiomedical recorder.,” *Hypertension*, vol. 27, no. 6, pp. 1318–24, 1996.

[23] M. Kato, B. G. Phillips, G. Sigurdsson, K. Narkiewicz, C. A. Pesek, and V. K. Somers, “Effects of sleep deprivation on neural circulatory control,” *Hypertension*, vol. 35, no. 5, pp. 1173–1175, 2000.

[24] H. K. Meier-Ewert *et al.*, “Effect of sleep loss on C-Reactive protein, an inflammatory marker of cardiovascular risk,” *J. Am. Coll. Cardiol.*, 2004.

[25] B. Rasch and J. Born, “About sleep’s role in memory.,” *Physiol. Rev.*, 2013.

[26] M. Abo-Zahhad, S. M. Ahmed, and S. N. Abbas, “A New EEG Acquisition Protocol for Biometric Identification Using Eye Blinking Signals,” *Int. J. Intell. Syst. Appl.*, vol. 7, no. 6, pp. 48–54, 2015.

[27] M. A. Carskadon and W. C. Dement, “Normal Human Sleep : An Overview,” *Princ. Pract. sleep Med.*, pp. 16–26, 2011.

[28] P. Chriskos *et al.*, “Automatic Sleep Stage Classification Applying Machine Learning Algorithms on EEG Recordings,” *2017 IEEE 30th Int. Symp. Comput. Med. Syst.*, pp. 435–439, 2017.

[29] D.-J. Dijk and S. W. Lockley, “Integration of human sleep-wake regulation and circadian rhythmicity.,” *J. Appl. Physiol.*, vol. 92, no. 2, pp. 852–862, 2002.

[30] HOWSLEEPWORKS, “Sleep Cycle – Types and Stages of Sleep,” 2017. [Online]. Available: https://www.howsleepworks.com/types\_cycles.html.

[31] E. Alpaydın, *Introduction to machine learning*, vol. 1107. 2014.

[32] B. Hjorth, “EEG analysis based on time domain properties,” *Electroencephalogr. Clin. Neurophysiol.*, 1970.

[33] R. Polikar, “The Wavelet Tutorial,” *Internet Resour. httpengineering rowan edu polikarWAVELETSWTtutorial html*, 1994.

[34] P. Achermann, “EEG Analysis Applied to Sleep,” *Sleep*, 2009.

[35] P. D. Welch, “The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms,” *IEEE Trans. Audio Electroacoust.*, 1967.

[36] S. M. Kay, “Modern Spectral Estimation: Theory and Application,” in *Signal Processing. Prentice-Hall*, 1988.

[37] S. Haykin, *Adaptive Filter Theory*. 2002.

[38] S. M. Pincus, I. M. Gladstone, and R. A. Ehrenkranz, “A regularity statistic for medical data analysis,” *J. Clin. Monit.*, 1991.

[39] A. Dauphiné, *Fractal Geography*. 2013.

[40] J. Theiler, “Estimating fractal dimension,” *J. Opt. Soc. Am. A*, 1990.

[41] I. Steinwart and A. Christmann, *Support Vector Machines*, vol. 13, no. 4. 2010.

[42] DTREG, “SVM - Support Vector Machines.” [Online]. Available: https://www.dtreg.com/solution/view/20.

[43] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “kNN Model-Based Approach in Classification,” *Move to Meaningful Internet Syst. 2003 CoopIS, DOA, ODBASE*, vol. 2888, pp. 986–996, 2003.

[44] L. Rokach and O. Maimon, “Decision Trees,” in *Data Mining and Knowledge Discovery Handbook*, 2010, pp. 165–192.

APPENDIX A

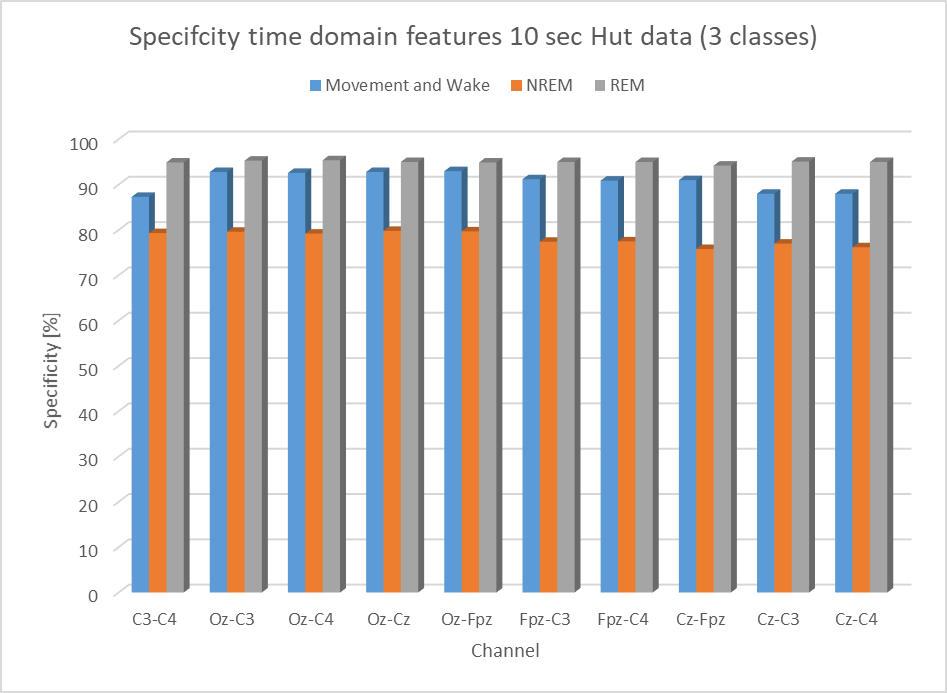
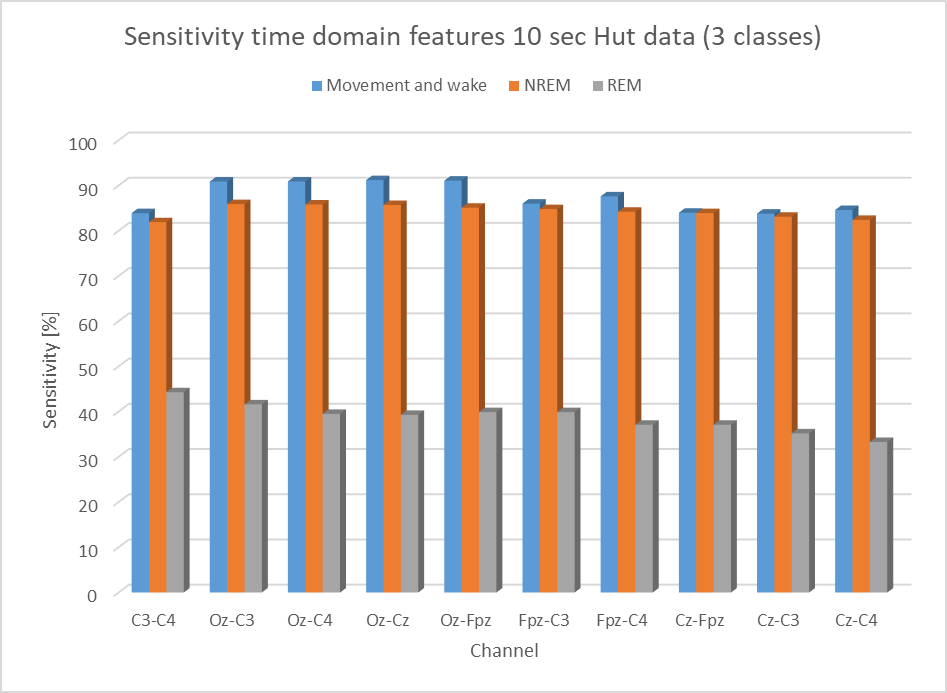
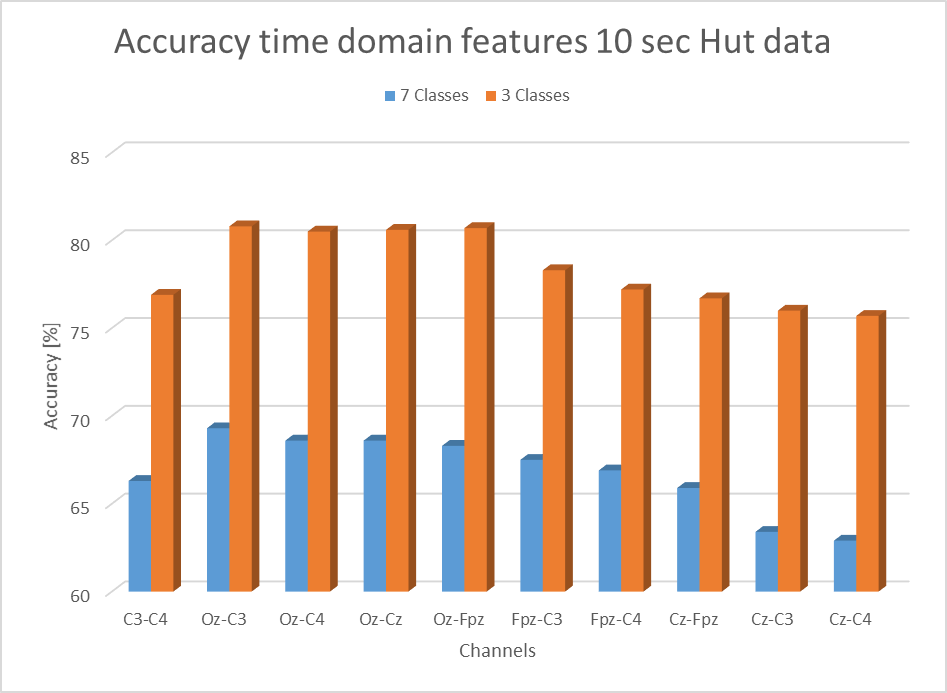
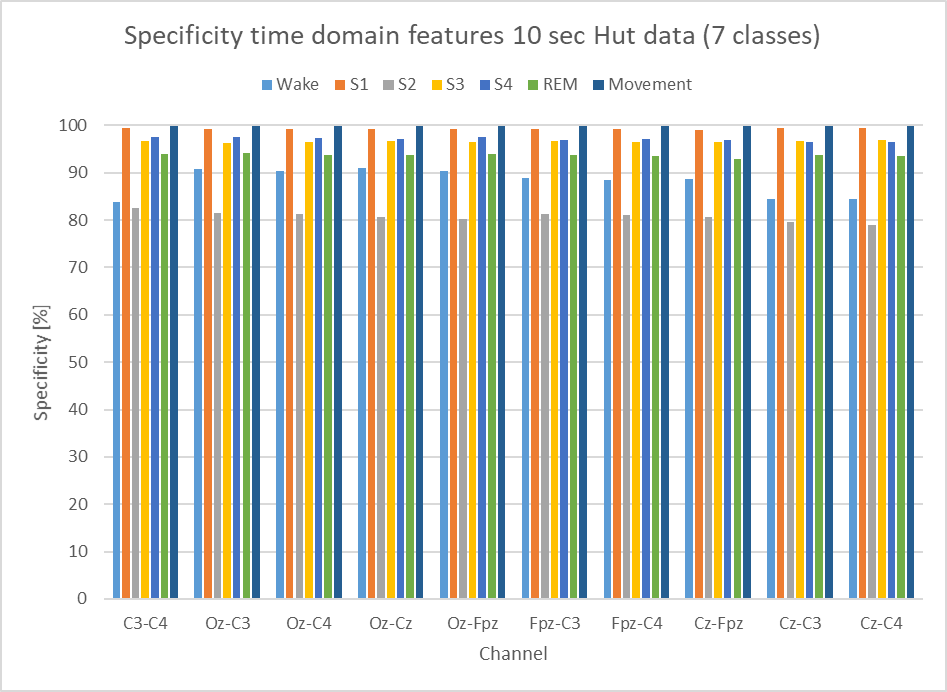
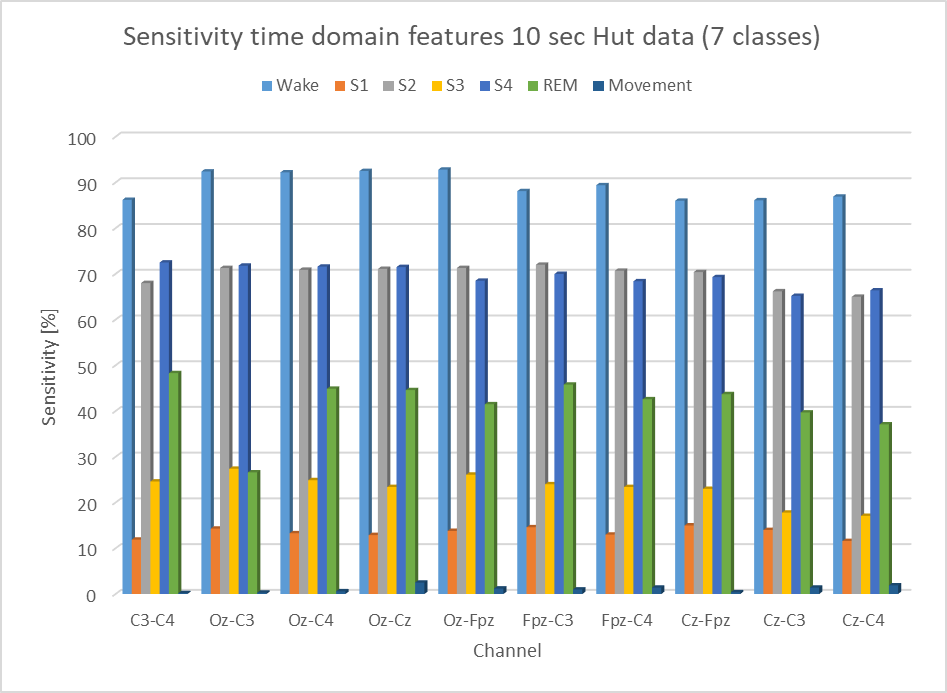


Figure 42 Characteristics of the RF model performance under the time domain features.

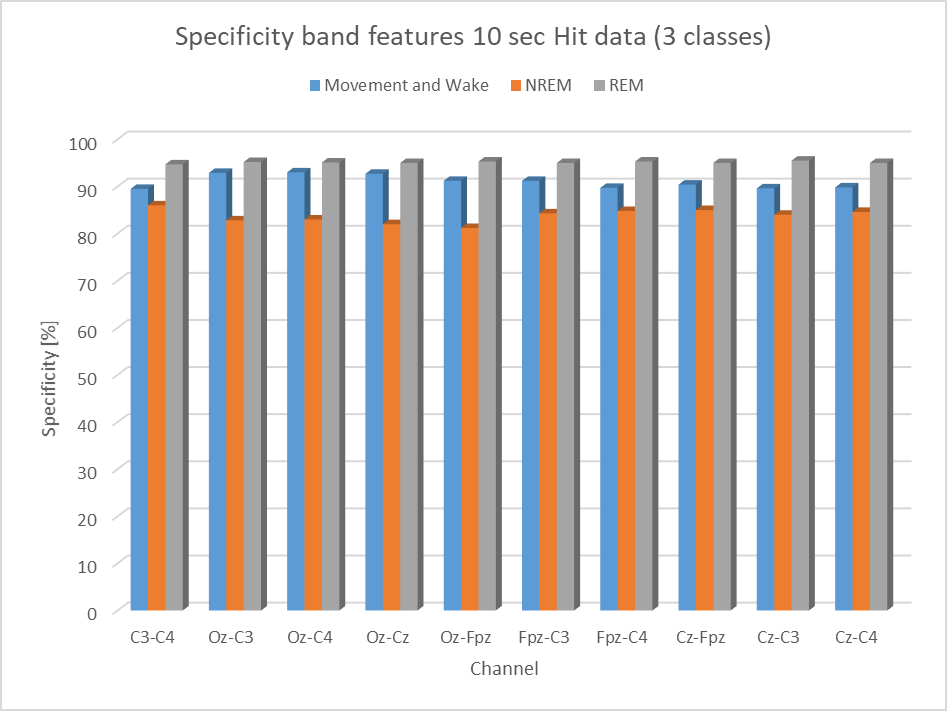
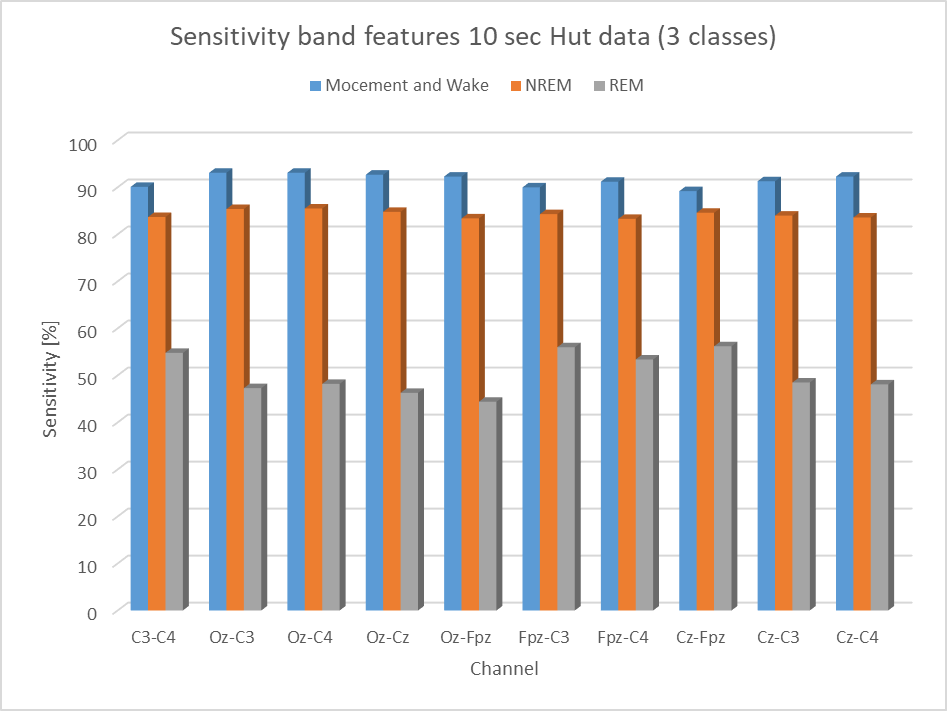
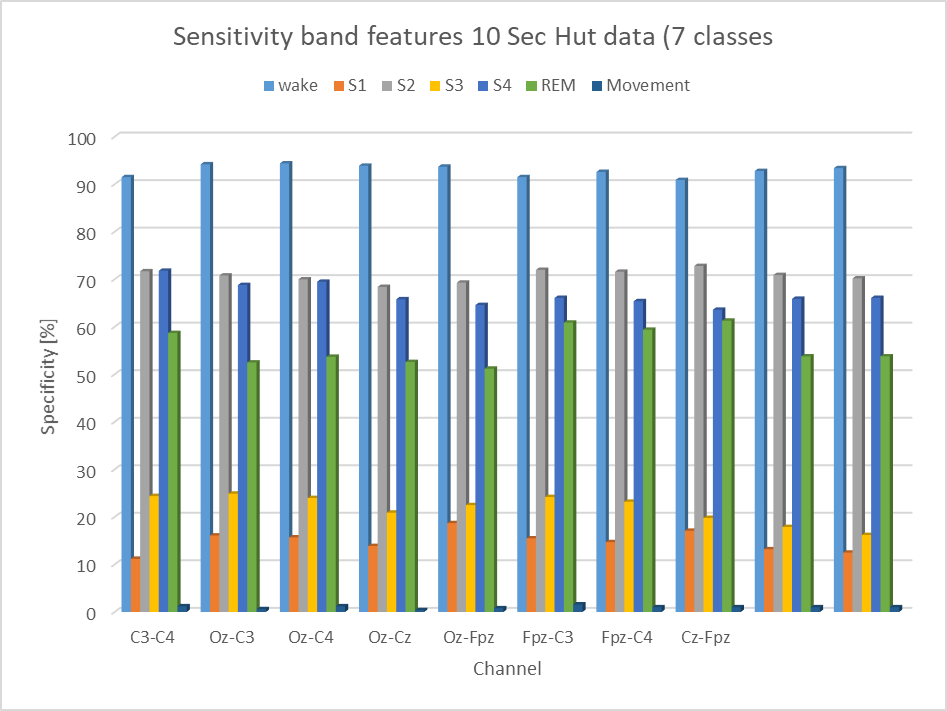
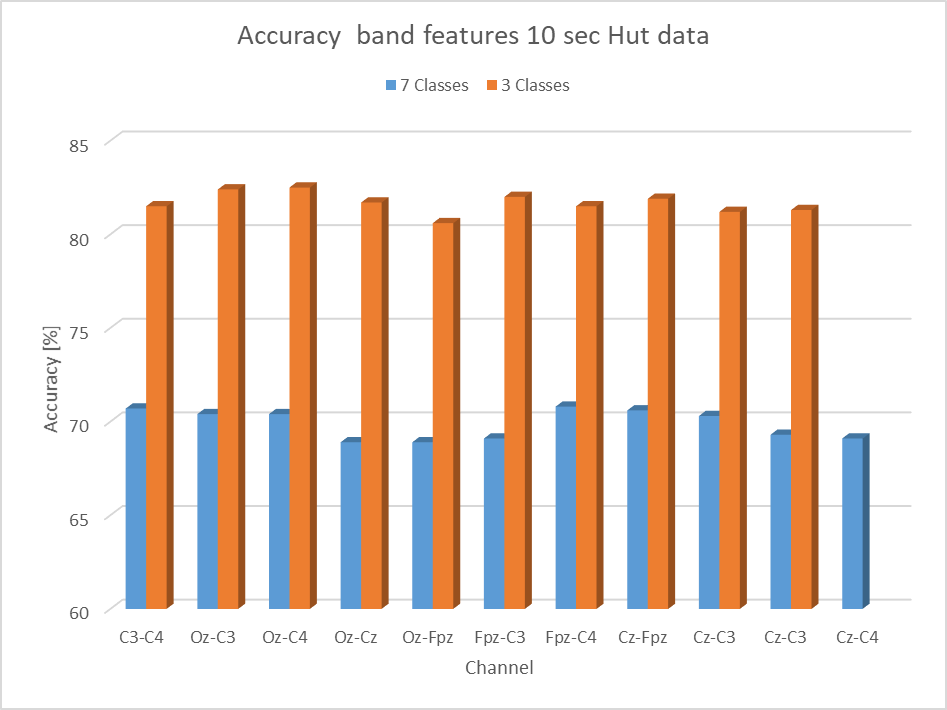


Figure 43 Characteristics of the RF model performance under the frequency domain band features

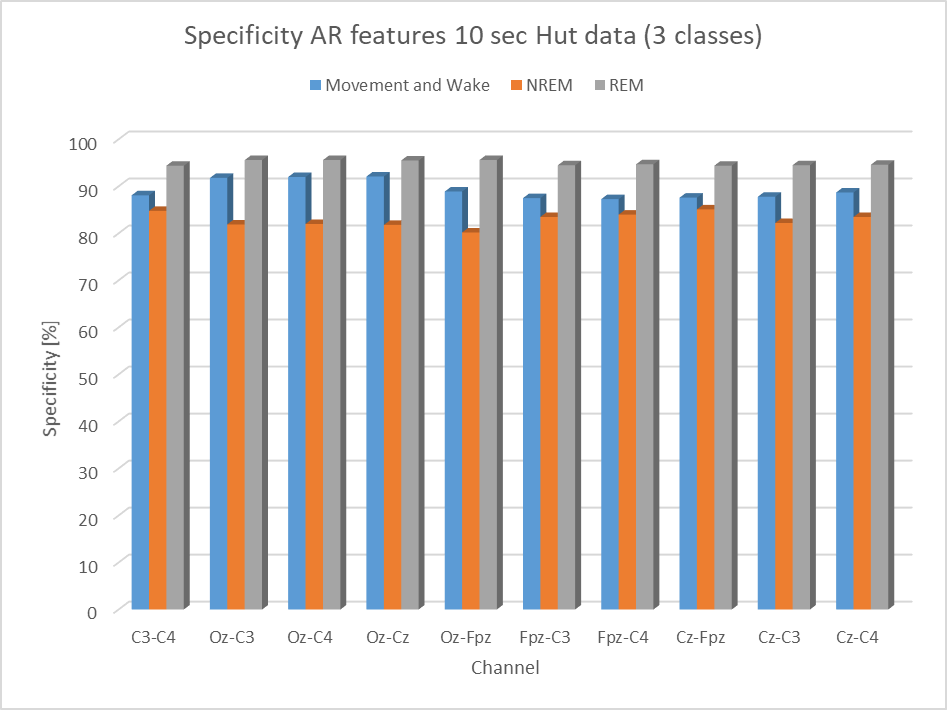
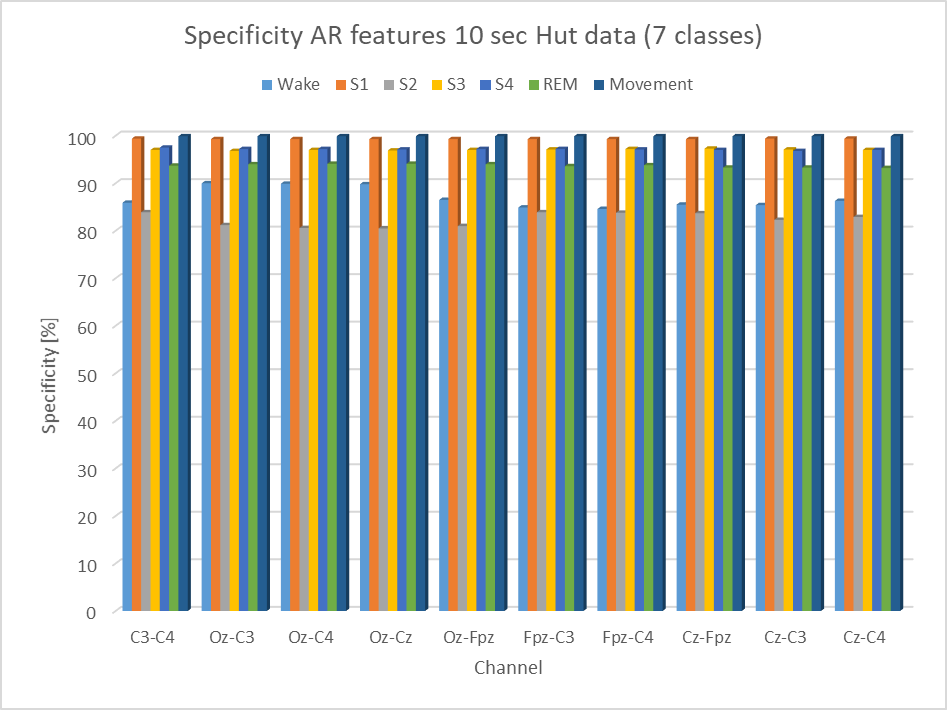
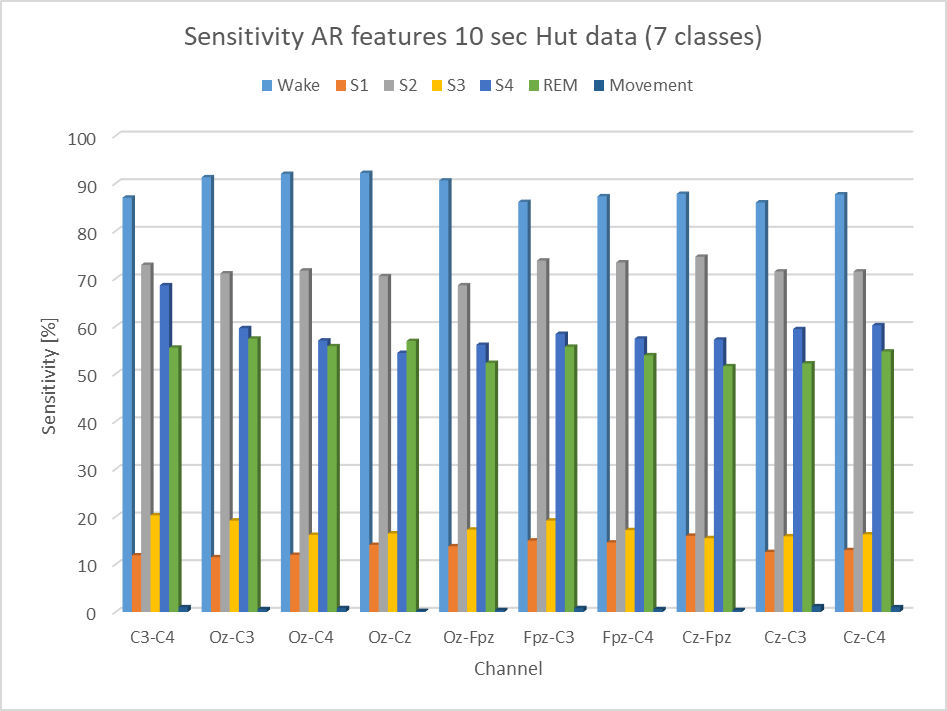
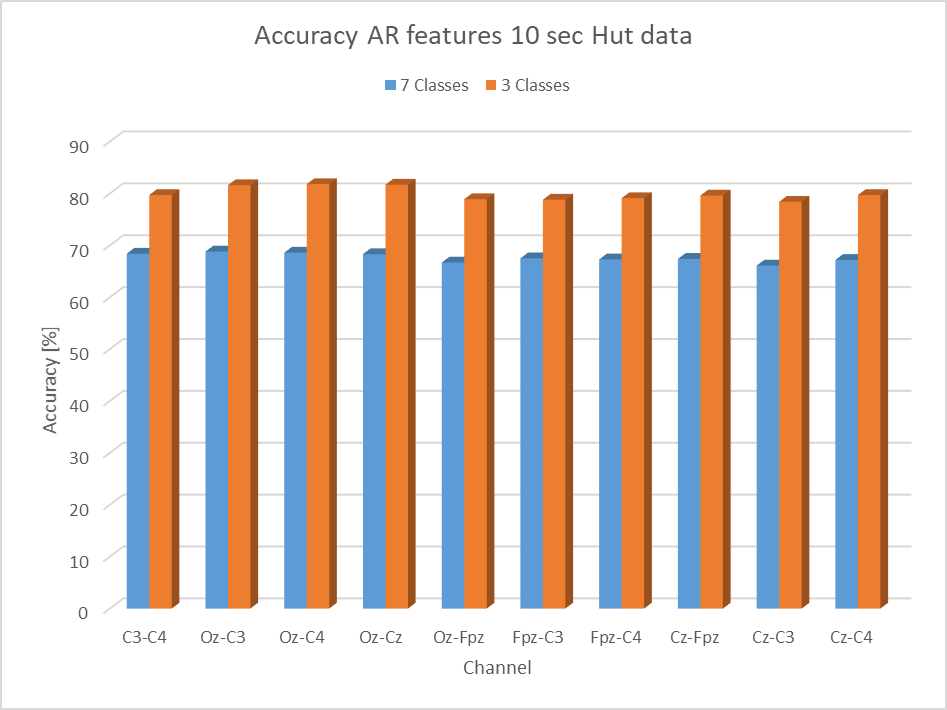
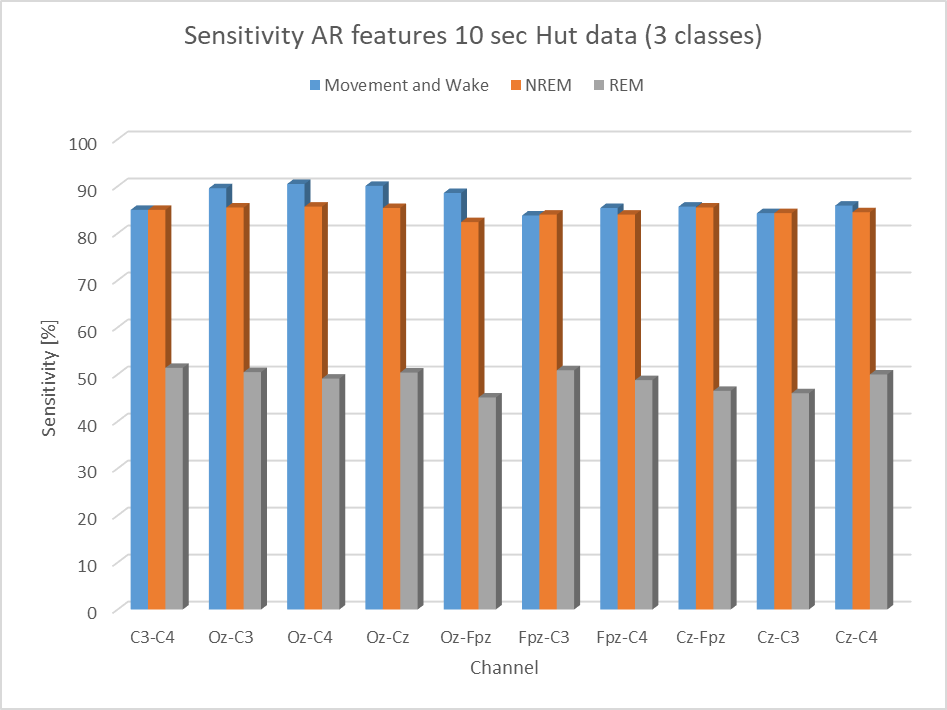


Figure 44 Characteristics of the RF model performance under the frequency domain AR features.

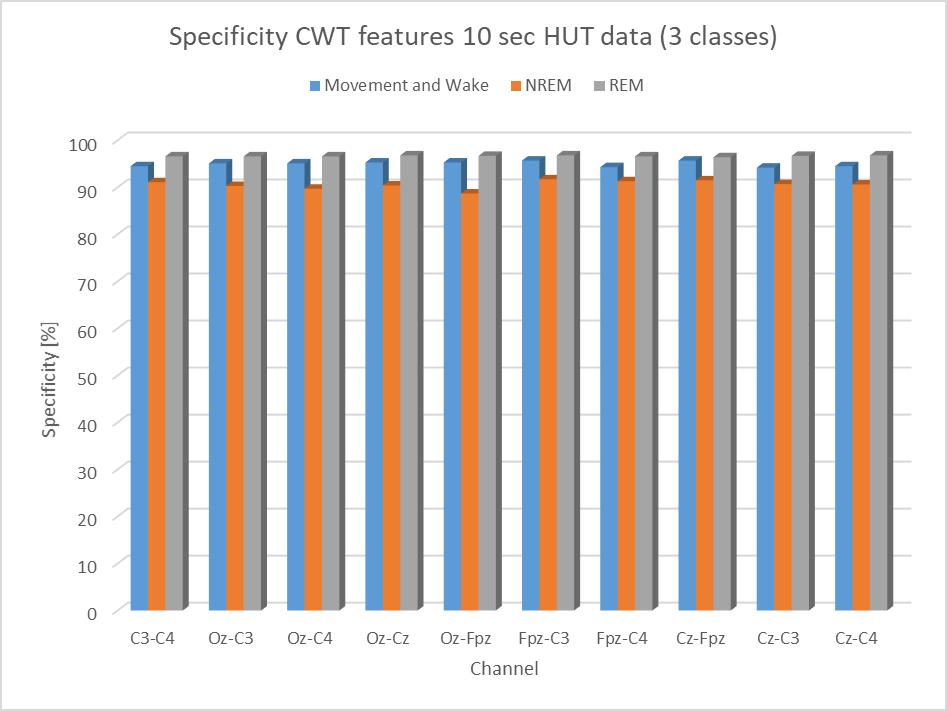
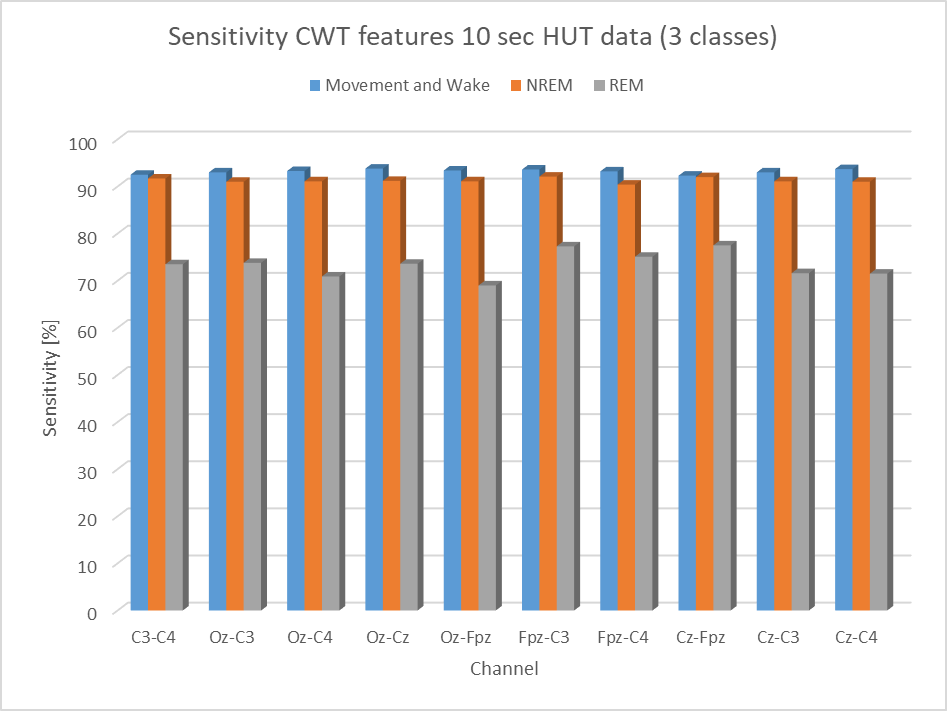
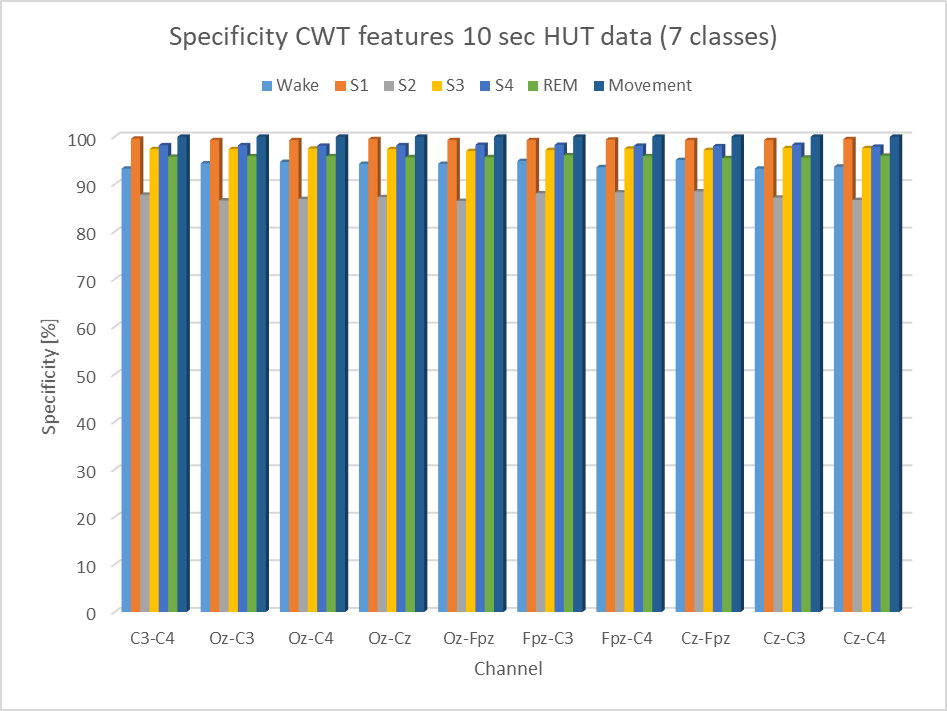
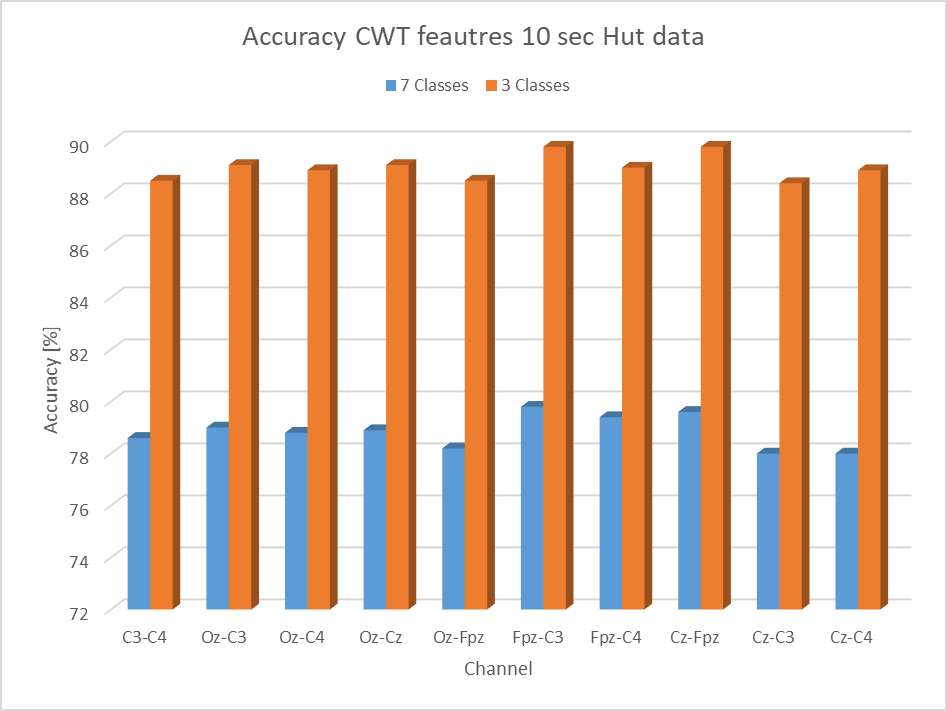


Figure 45. Characteristics of the RF model performance under the continuous wavelet features