Hanzehogeschool Groningen

# 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

Sleeping is an important part of human life as multiple sleep restriction studies have shown. However, its function is not yet completely understood even though researchers have been performing sleep experiments for years. These experiments require sleep stage scoring, which as of this moment is done visually by trained experts. However, this method has multiple drawbacks. In order to attempt to resolve these challenges, an automatic sleep scoring procedure is proposed in this document. Using Random Forest and K-nearest neighbor classifier algorithms, 4 separate models, for a 3 class (Wake, NREM, and REM) and a 6 class (Wake, S1, S2, S3, S4, REM) each, were trained. These models were trained on several feature sets: time domain features, Autoregressive (AR) model coefficient features, and Continuous Wavelet Transform (CWT) spectral band features. Furthermore, a Hidden Markov Model (HMM) was trained on a combined feature set where each sleep stage represented a hidden state. This allowed for a comparison between unsupervised and supervised algorithms based on the same feature set. Finally, an overall comparison of the methods concludes that the 3 class RF model based on the CWT feature set had the highest accuracy of 88.8% under the Cz-Fpz EEG channel.

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

I would like to thank my technical project supervisor Prof. Dr. Roelof Hut, for his guidance for the whole duration of the project. Without him, the project would not have been possible. His continued interest in the topic and his out-of-the-box thinking are what fueled the progress of this study.

I would also like to thank my company supervisor, Ir Charissa Roossien. Her patience, encouragement and meaningful critique are what let me achieve the goals of this project.

I would also like to thank all my colleagues at the Chronobiology department at the RUG for the pleasant workplace they provided and their help

Finally, I would like to thank my family and friends for their continued support.

TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# 1. 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. An automated system could 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?

# 2. 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 2.1 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.

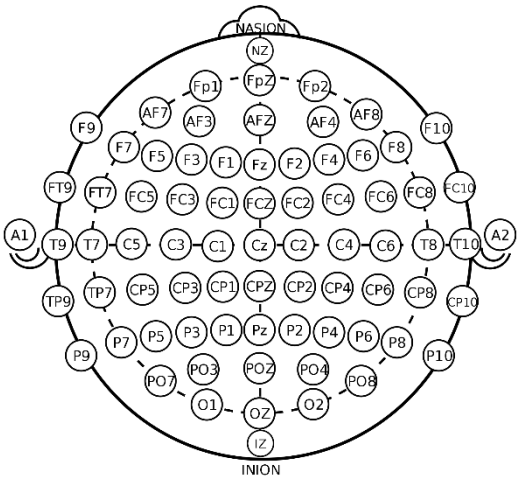



*Figure 2.1 The expanded 10-20 system for EEG electrode placement*

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 2.1.

*Table 2.1 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 2.2 and 2.3.

- 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)

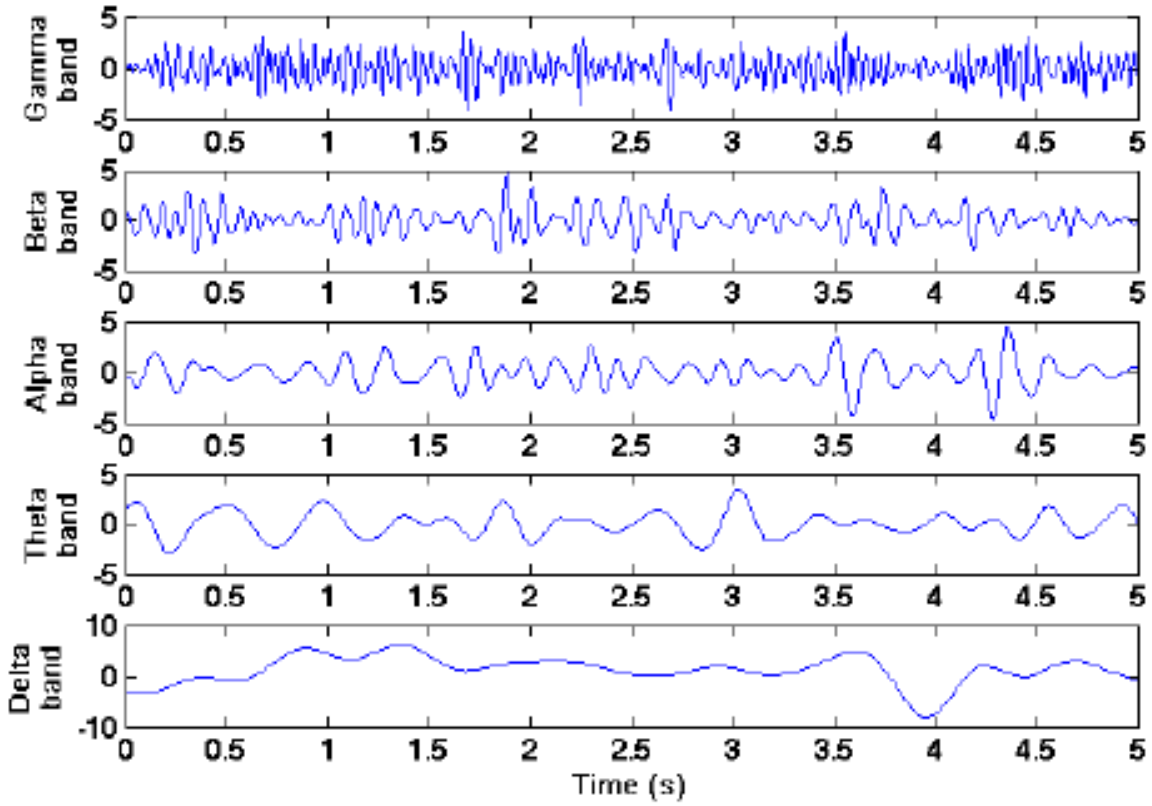



*Figure 2.2. Frequency bands of the sleep EEG signal represented as signals in time.* [26]

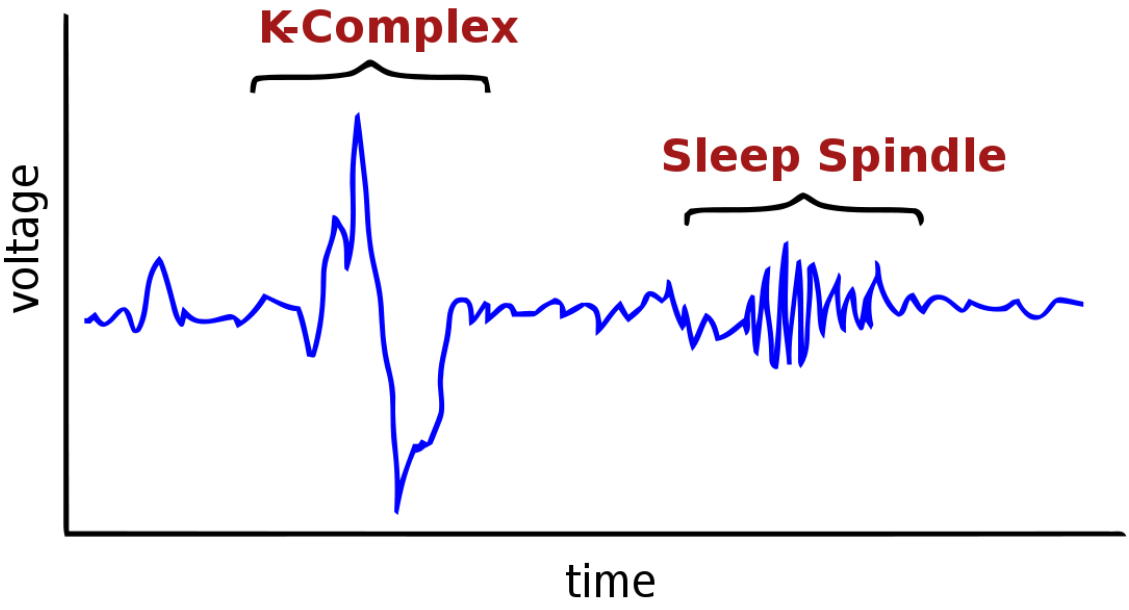



*Figure 2.3. K-complexes and Sleep spindles as parts of an EEG signal in time*

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 2.4 in which the separate cycles of sleep are shown as a function of time during an 8-hour long night sleep.

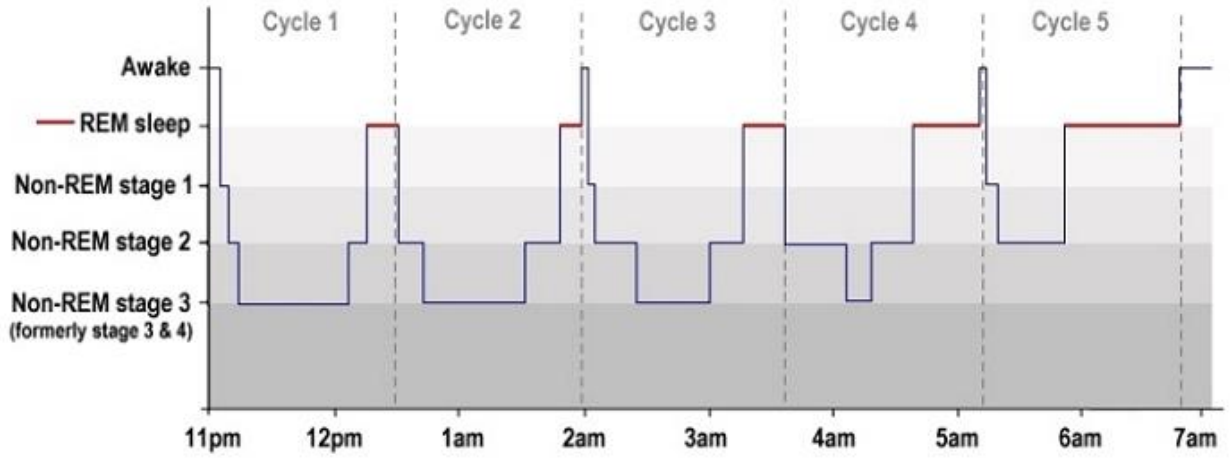



*Figure 2.4. A hypnogram showing sleep stages and cycles in time* [30]

## 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 2.5 [31]. Therefore, feature extraction is a critical step in building any automatic classification system.

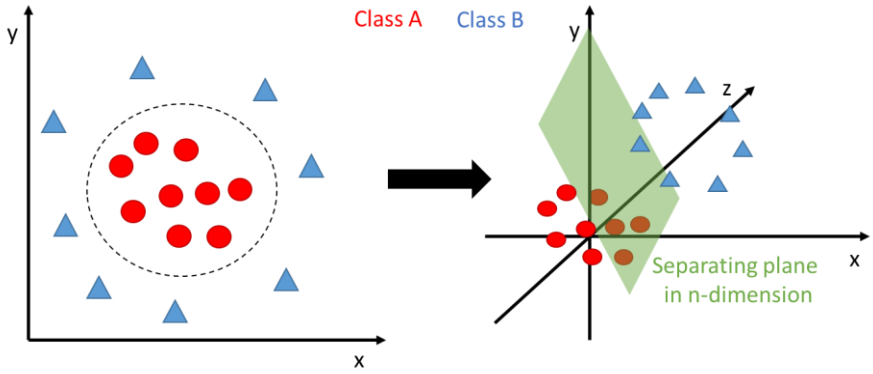



*Figure 2.5. Data visualisation under different feature spaces. Moving from a lower dimensional feature space (left) to a higher dimensional feature space (right) can make it easier to segregate the classes.*

### 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 available data is a time-series, meaning a signal composed of data points recorded in time. As previously 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 this raw EEG epoch data is fed 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 it should be noted that this is calculated by taking the minimum mentioned requirements. In fact, in this 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 and standard deviation,[10], [12] given by the following formulas:

$$Mean = \frac{\sum x}{N} \quad (1)$$

$$Standard\ Deviation = \sqrt{\frac{\sum (x - Mean)^2}{N - 1}} \quad (2)$$

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

**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 2.6:

$$ZCR = \frac{1}{T-1}\sum_{t=1}^{T-1} 1_{R<0}(x_t x_{t-1}) \quad (5)$$

Where x is the signal of length T and $1_{R<1}$ is an indicator function given by:

$$1_A(x) = \begin{cases} 1 & \text{if x} \in \text{ A,} \\ 0 & \text{if } x \notin A. \end{cases}$$

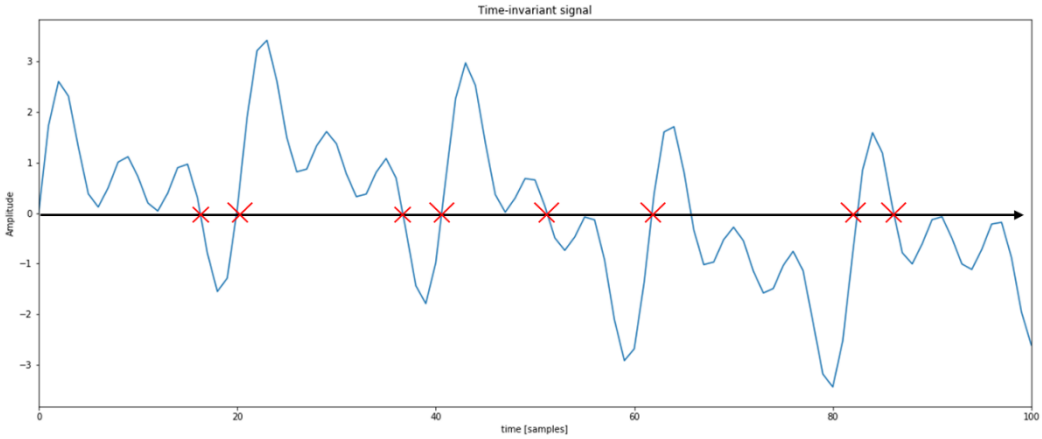



*Figure 2.6. The zero-crossing rate of a time-invariant signal. Notice how the number of zero-crossings is not representative of the frequency of the signal*

**Hjorth parameters**

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]:

$$Activity = var\big(y(t)\big) \quad (6)$$

Where var is given by:

$$var\big(y(t)\big) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)^2 \quad (7)$$

Where μ is the mean.

$$Mobility = \sqrt{\frac{var(\frac{dy(t)}{dt})}{var(y(t))}} \quad (8)$$

$$Complexity = \frac{Mobility(\frac{dy(t)}{dt})}{Mobility(y(t))} \quad (9)$$

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 it can be seen 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**

The topic of extracting the frequencies from the time series has already been reviewed to an extent. 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 yields is the intensity of every frequency present in the time series. Since the EEG signal is a time-series with certain sampling frequency the discrete Fourier transform (DFT)[33] given by the following formula can be used:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi i}{N} kn} \qquad (10)$$

Where $X_k$ is the transformed data point at location ***k***, ***N*** is the number of data points and $x_n$ 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 this purpose as seen in figure 2.7. 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].

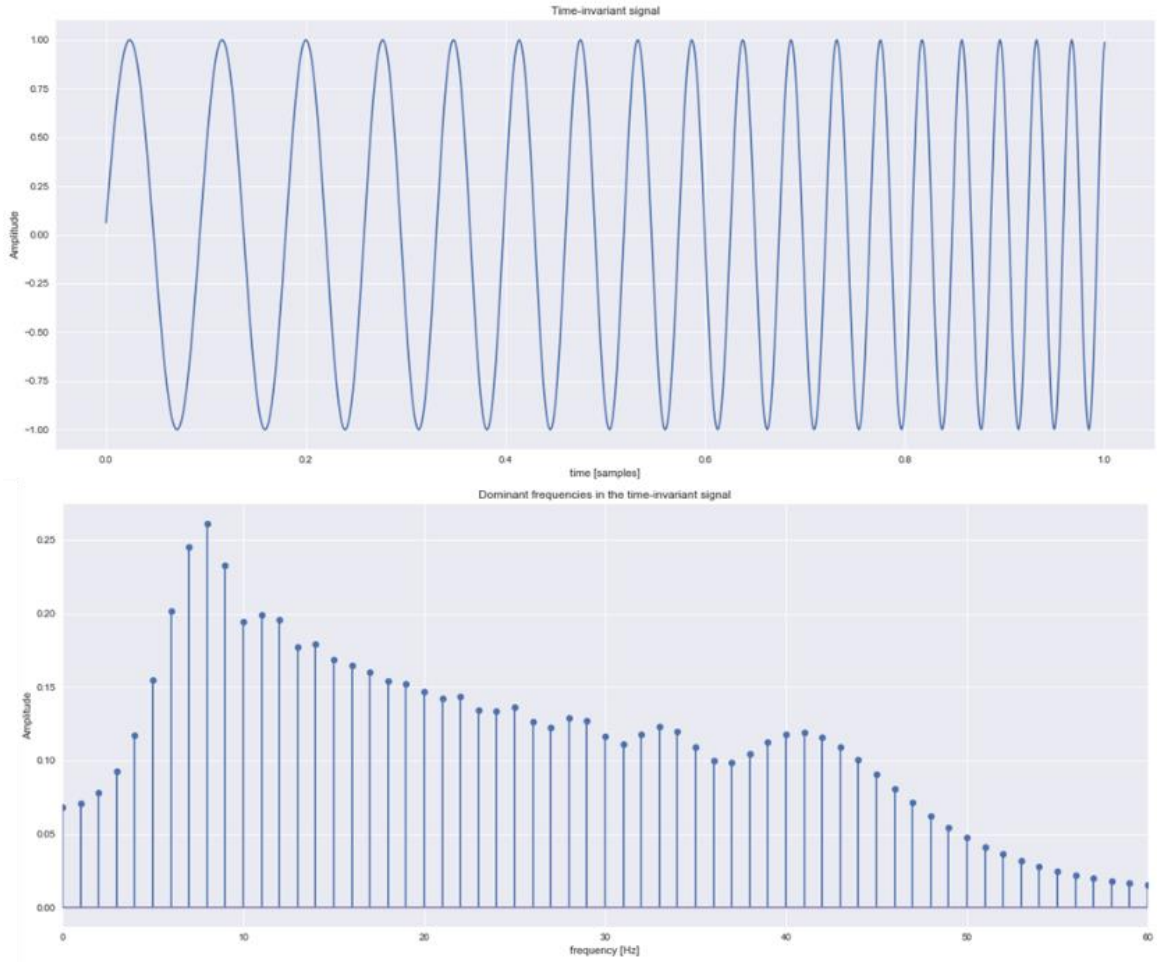



*Figure 2.7 A time varying signal and its DFT magnitude plot.*

If the sample size is not representative the DFT would yield a magnitude plot with an overwhelmingly dominant $0^{th}$ 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 2.8.

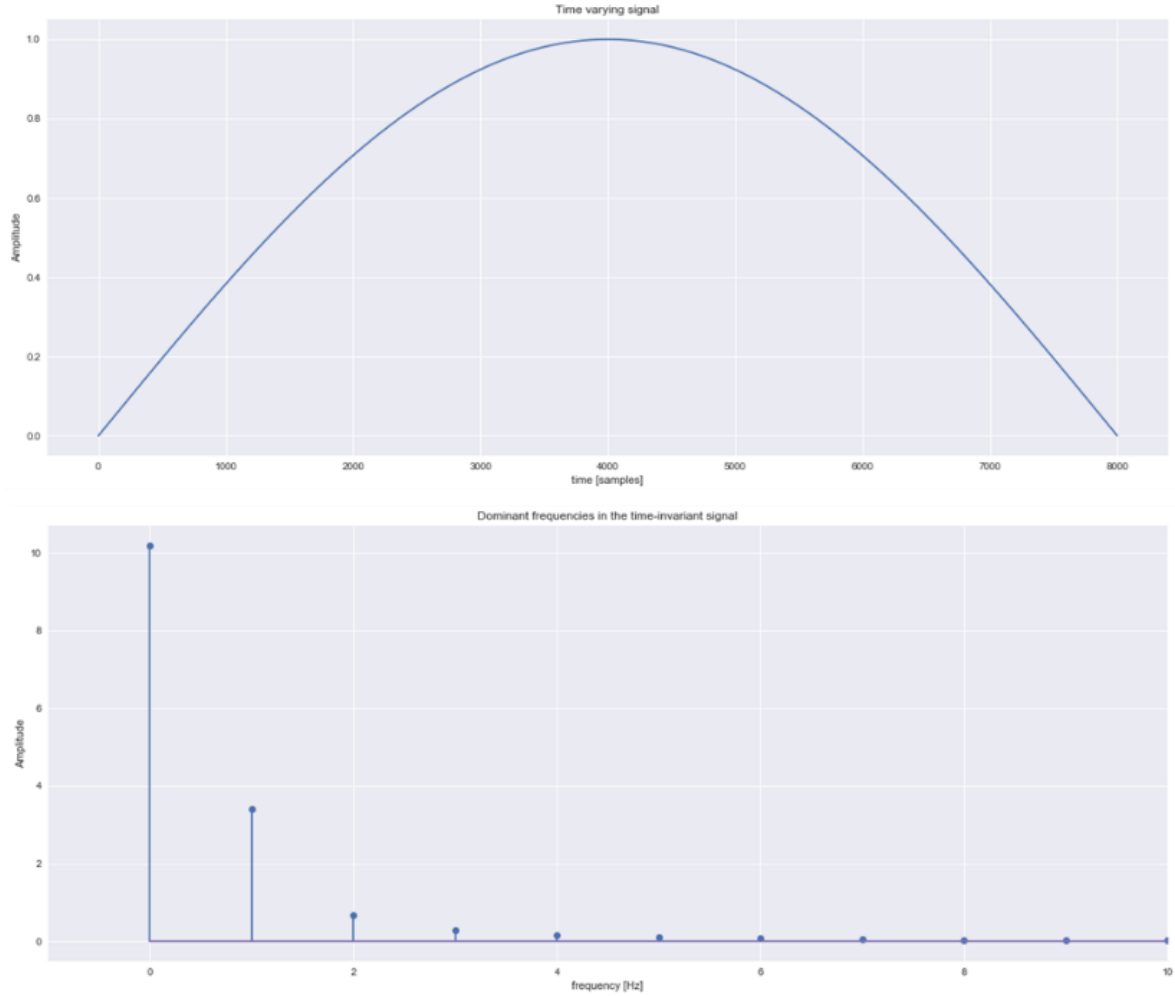



*Figure 2.8 Half a sine wave and its DFT magnitude plot.*

In this 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 this 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 this system [5].

**Autoregressive model coefficients (Parametric)**

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 the EEG signals can be modelled by an AR process [35], [36].

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 it is possible to 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 [36]:

$$X_t = \varphi_0 + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \quad (11)$$

$$X_t = \varphi_1 X_{t-1} + \cdots + \varphi_p X_{t-p} + \varepsilon_t \quad (12)$$

Where an AR(p) is an autoregressive model of order **p,** $\boldsymbol{\varphi_1}$, … , $\boldsymbol{\varphi_p}$ are the coefficients (weights) of the model, $\boldsymbol{\varphi_0}$ is a constant and $\boldsymbol{\varepsilon_t}$ is white noise. By looking at the formula, it can concluded that the variable which provides 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 [36]:

$$P_x(\omega) = \frac{\sigma_w^2}{\left|1 + \sum_{i=1}^{p} \varphi_i e^{-j\omega i}\right|^2} \quad (13)$$

Where $\boldsymbol{\sigma_w^2}$ 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, they can used as a characteristic for the PSD of the signal model. It is important to emphasize that the PSDs extracted from the epochs used in the non-parametric method from section 2.2.2.2 and the ones simulated with the AR model should theoretically provide almost identical magnitude plots as seen in figure 2.9. Therefore, it will be redundant to repeat the step of separating the spectrum into bands and integrate the data. Instead, the coefficients themselves will be used as features for the automated scoring algorithm. It is reasonable to do that because no new perspective on the data is given if the same process from the non-parametric part is repeated. However, 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 2.10.

Since the coefficients will be used, a way to extract them from the raw data is needed. Fortunately, the Yule-Walker equations allow to easily do that. They can be expressed in the compact matrix form [36]:

$$Rw = r \quad (14)$$

Where *R* is the correlation matrix containing the calculated values from the autocorrelation function of the AR process, *w* are the weights given by $w_{i=\_}\varphi_i$ and *r* are the values for the correlation. Thus solving for *w* assuming that R is nonsingular (can be inverted) from the compact matrix form as follows [36]:

$$w = R^{-1}r \quad (15)$$

Which in expanded form would look as follows [36]:

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} r(0) & r(1) & \dots & r(n-1) \\ r^*(1) & r(0) & \dots & r(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(n-1) & r^*(n-2) & \dots & r(0) \end{bmatrix}^{-1} \begin{bmatrix} r^*(1) \\ r^*(2) \\ \vdots \\ r^*(n) \end{bmatrix} \quad (16)$$

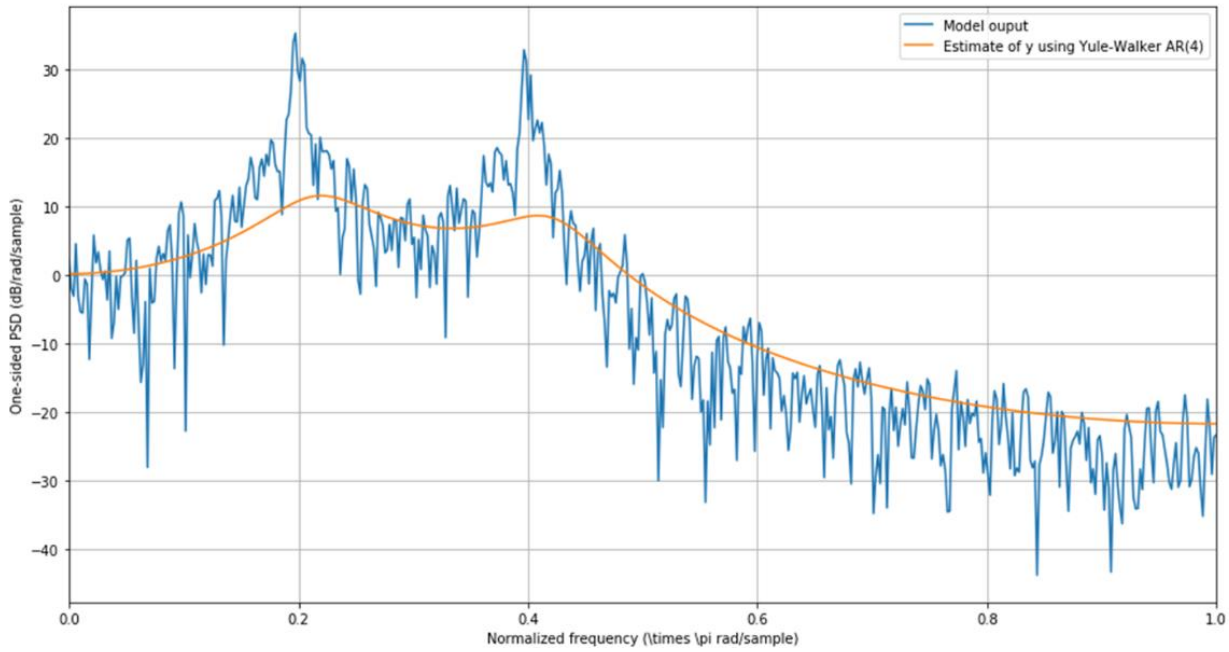



*Figure 2.9. A generated AR model PSD output (blue) and its estimation based on the coefficients (n=4) extracted by using the Yule-Walker equations*

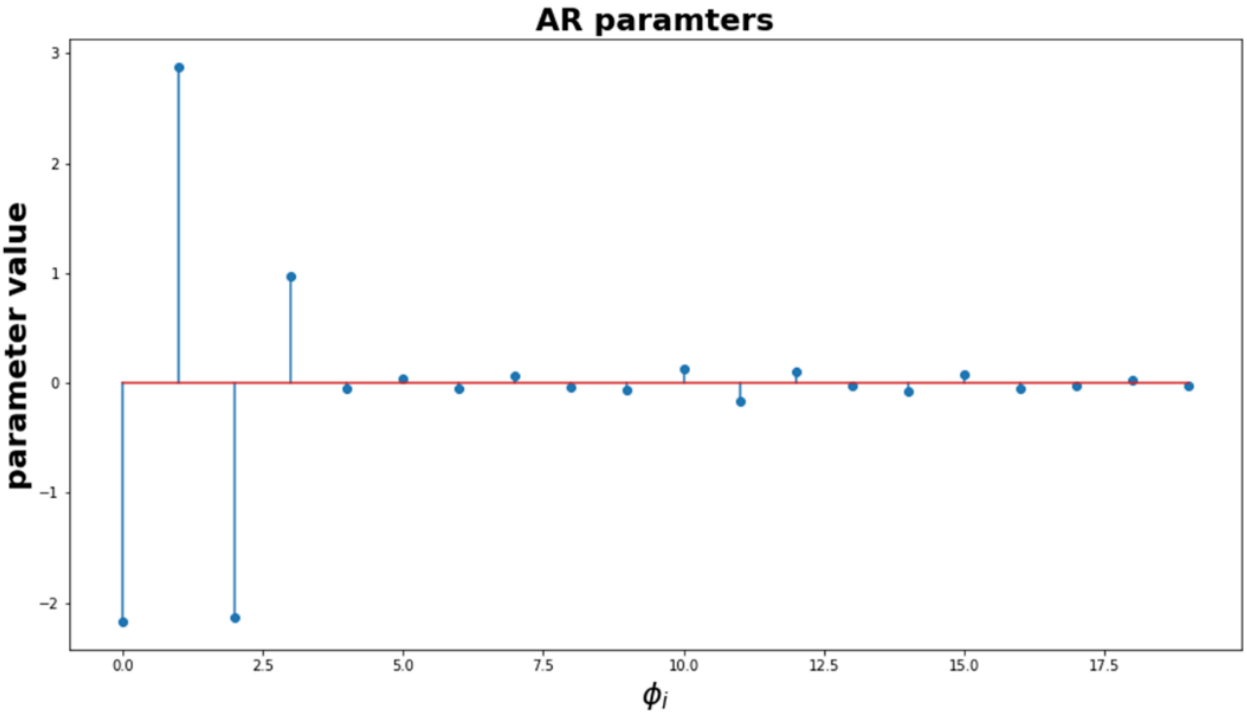



*Figure 2.10. Evolution of the AR parameters with increasing **i** term*

#### 2.2.2.3 Time-frequency domain

**Time-varying signals**

While the PSD representation provides 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. Perhaps, greater accuracy would be gained from an ML algorithm using a

representation which accounts for both of these issues. Therefore, a signal representation in both the time and frequency domain simultaneously might be beneficial [33].

### The Heisenberg uncertainty principle

Before discussing the time-frequency domain representation of the 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 this case, it forces a decision of which one is preferred: the time resolution or the frequency resolution.

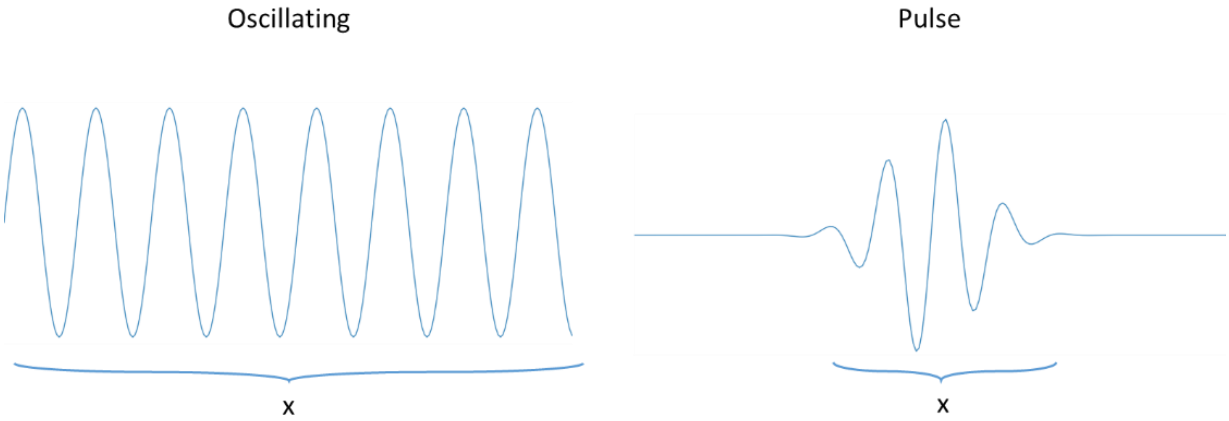



*Figure 2.11. Visualization of the uncertainty principle. On the left, the oscillating wave presents more information about the frequency (momentum) of the wave (particle) and insufficient information about its position, and on the right, the pulse wave presents much more information about the position of the particle but less about the frequency.*

### Short-Time Fourier transform (STFT)

As already discussed, due to TUP the exact time-frequency information of a time-varying signal cannot be extracted. Fortunately, in this 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 provides both time and frequency information simultaneously, even if it is not exact. The first technique to do that is the discrete-time STFT defined as [35]:

$$X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \quad (17)$$

Where x[n] is the signal and w[n] is a window function. This window function is the source of an important limitation for the STFT, presented in the form of spectral leakage. Additionally, the decision of having to choose either a good frequency or time resolution comes with the limitation of having to linearize both, meaning that the scale for the bins in both domains are linear [35]. Even though the STFT would provide a representation of the signal both in the time and the frequency domain, the limitations it brings lead to the conclusion that another method might be more suitable.

**Continuous Wavelet Transform**

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 in this case. A technique called the Continuous Wavelet Transform (CWT) allows something similar [33].

The CWT is a technique for time-frequency analysis which ultimately compares two signals: the original signal to be analyzed and the generated one called the 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 for gaining 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 a 2D representation of the signal can be created where one dimension is given by the scale of the wavelet and the other by the time shifts [33]. The process is illustrated in figure 2.12.

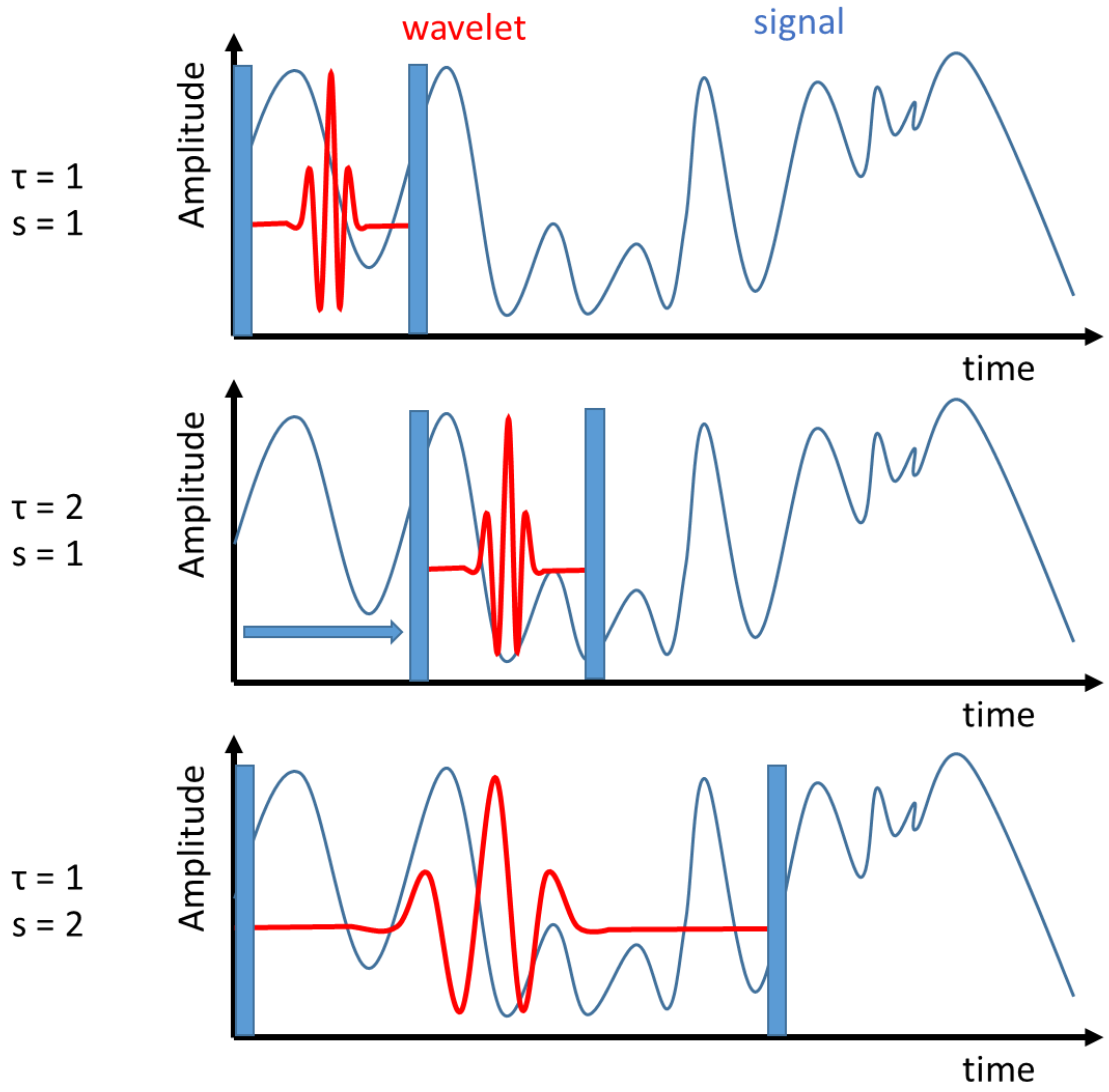



*Figure 2.12. A visual representation of how the CWT algorithm works where* **t** *indicates time step and* **s** *indicates scale. First, a low scale is selected (compressed mother wavelet) and it is being time-shifted as to be compared to every temporal bin of the signal. Afterward, a new scale is selected and the process repeated.*

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

$$X_{\omega}(s,\tau) = \frac{1}{|s|^{1/2}} \int_{-\infty}^{\infty} x(t)\varphi\left(\frac{t-\tau}{s}\right) dt \quad (18)$$

where φ is the complex conjugate of the mother wavelet signal represented in both the time and frequency domain. As it can be seen 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 it allows for a non-linear representation as seen in figure 2.13. While the STFT provides bins of equal width and length, the ones from the CWT are distributed unevenly which is exactly the characteristic needed for this case [33].

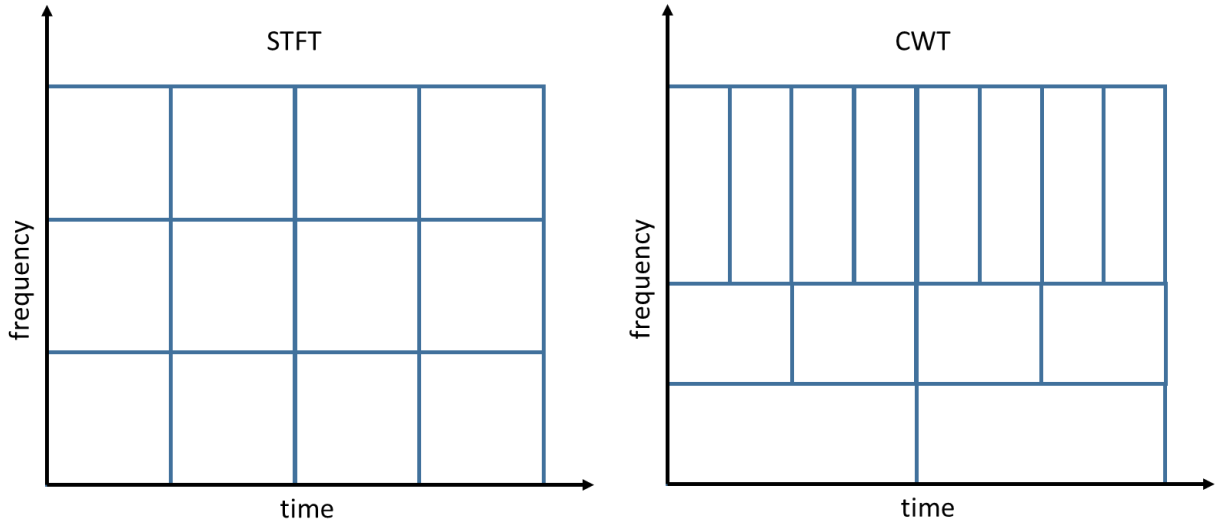



*Figure 2.13. An example bin representation of the spectrograms of STFT (left) and CWT₃ (right). It is apparent how the STFT has a linear distribution while the CWT allows for higher resolution at specific areas*

### 2.2.3 Classifier types

**KNN**

The K-nearest neighbor (KNN) method requires a feature space in which to operate and classifies the data point based on a majority vote from its neighbors. KNN is instance based and the class determined locally [37], [38]. The is visualized in figure 2.14 where it can be seen how changing the number of neighbors ***k*** might result in a different majority vote and therefore class assignment.

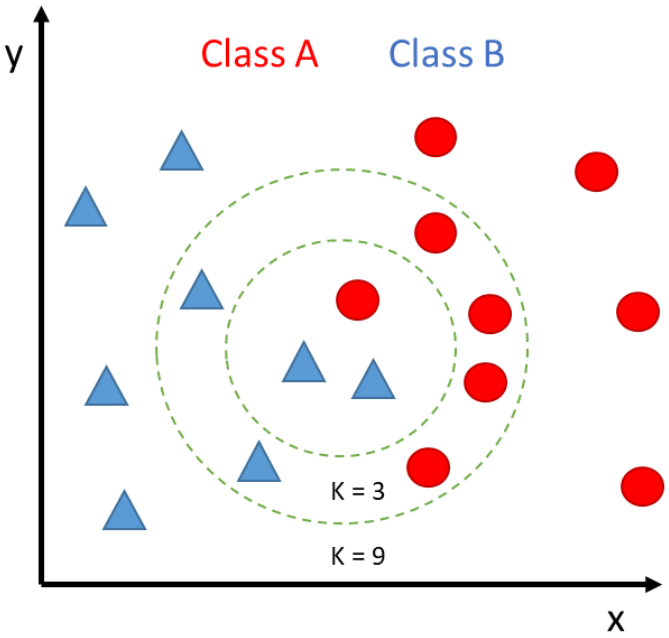



*Figure 2.14. KNN algorithm under varying values k fro number of neighbors and the consequences of these changes*

## 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 [39]. The workings can be seen in figure 2.15.

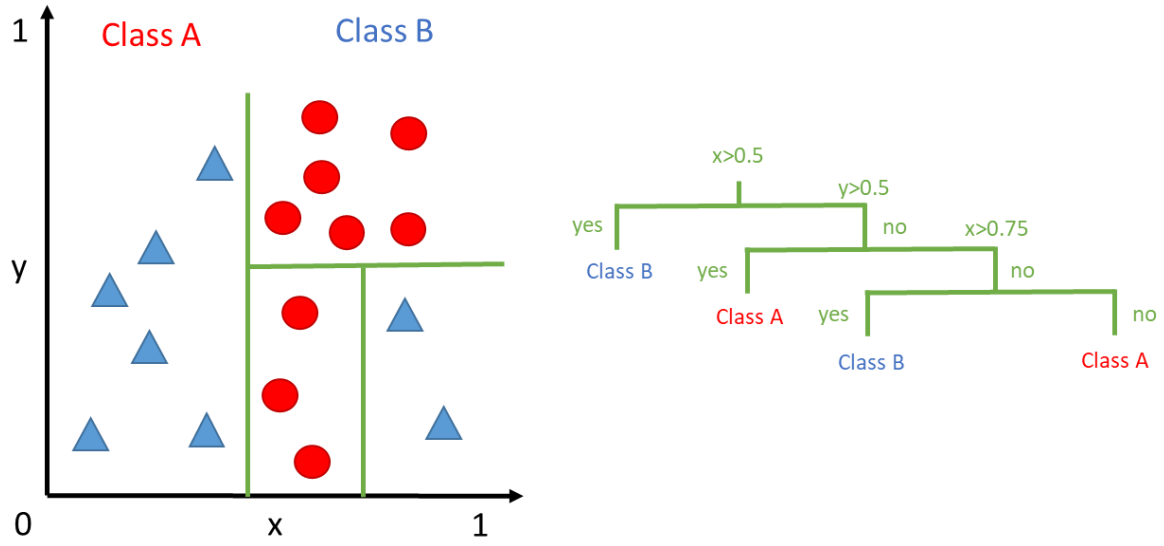
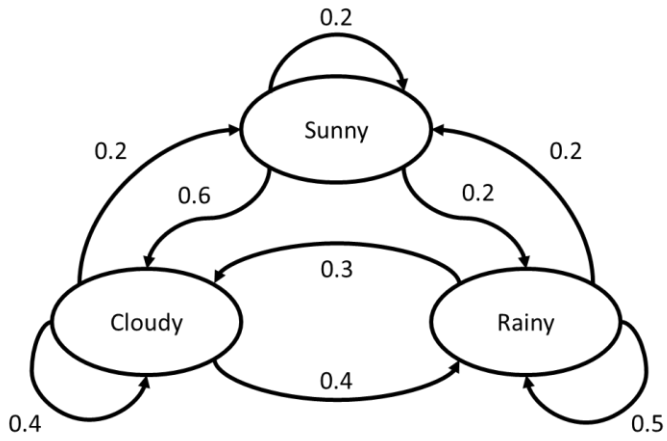



*Figure 2.15. Decision tree schematic where the circles represent a single binary condition.*

## 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 it can be assumed that the EEG signal is the result of a stochastic process [34], [36]. In this case, there is a finite set of observations (EEG epochs) and it can be assumed that a random state switching process exists, with the states being the sleep stages. Fortunately, these are all the preconditions required to train a Hidden Markov Model.

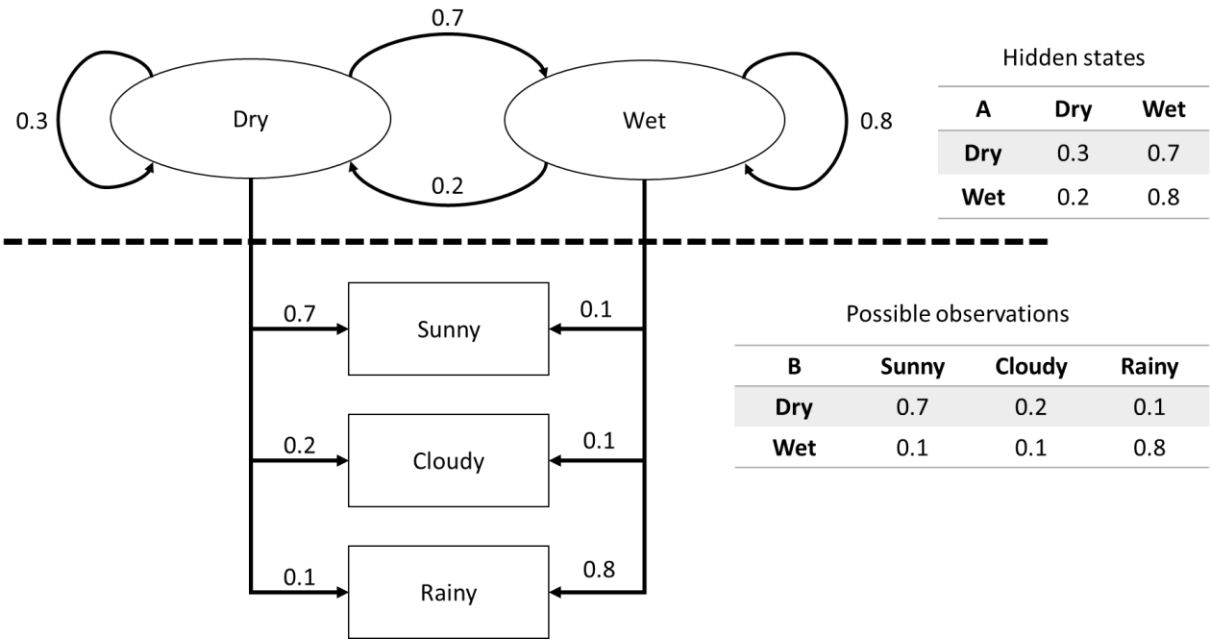
Markov chains are used when there is a system which can be described as being in one of several states at any time. A typical example is a simplified version of weather observation, where it can be assumed that the weather can be either sunny, cloudy or rainy presented in figure 2.16. There the probabilities of switching from one state to another can be seen given by matrix A. Then the possibility of having a certain sequence of observations occur can be calculated [14]:




| A | Sunny | Cloudy | Rainy |
|---|---|---|---|
| **Sunny** | 0.2 | 0.6 | 0.2 |
| **Cloudy** | 0.2 | 0.4 | 0.4 |
| **Rainy** | 0.2 | 0.3 | 0.5 |

*Figure 2.16. 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 this case, there is no 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, the situation presented in figure 2.17 occurs, where a hidden state switching process is yielding observations.




Hidden states

| A | Dry | Wet |
|---|---|---|
| Dry | 0.3 | 0.7 |
| Wet | 0.2 | 0.8 |

Possible observations

| B | Sunny | Cloudy | Rainy |
|---|---|---|---|
| Dry | 0.7 | 0.2 | 0.1 |
| Wet | 0.1 | 0.1 | 0.8 |

*Figure 2.17. 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 in figure 2.17 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].

$$(Initialization)\quad alpha_1(i) = \pi_i * b_i(X_1)$$

$$(Recursion)\quad alpha_1(j) = \left[\sum_{i=1}^{N} alpha_{t-1}(i) * a_{ji}\right] * b_j(X_t)$$

$$(Termination)\quad Probability = \sum_{i=1}^{N} alpha_T(i)$$

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 this case. It provides the most likely temporal evolution of the states, meaning which was the most likely state at each

observation. The Viterbi algorithm 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 it is possible to backtrack through the values in the additional variable and determine the most likely states. The algorithm is given by the following four steps [14]:

$$(Initialization)\ \ V_1(i) = \pi_i * b_i(X_1) \quad and \quad W_1(i) = 0$$

$$(Recursion)\ \ V_t(j) = \max_{1 \le i \le N} \left[V_{t-1}(i) * a_{ij}\right] * b_j(X_t) \quad and \quad W_t(j) = \operatorname{argmax}_{1 \le i \le N} \left[V_{t-1}(i) * a_{ij}\right]$$

$$(Termination)\ \ best\ score = \max_{1 \le i \le N} [V_T(i)] \quad and \quad S_t^* = \operatorname{argmax}_{1 \le i \le N} [V_T(i)]$$

$$(Backtracking)\ \ S_t^* = W_{t+1}(S_{t+1}^*) \quad for\ t = T-1, T-2, \dots, 1$$

$$S^* = (S_1^*, S_2^*, \dots, S_T^*)$$

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 would be solved if there is an already established HMM. However, in this case, there is no way of estimating the probabilities. Therefore, an algorithm which will estimate the model first is needed so that the Viterbi algorithm can be used 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, FA requires an already established HMM. So to start the BW either random values for the HMM parameters must be chosen or some approximate values assigned if there is previous knowledge about 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 the transition from one observation to the next in the observation sequence is included as well. In order to create a better estimate of the state transition matrix, two variables must be calculated. 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 shows how likely it is that the first state occurred 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 ksi variable has to be divided by the gamma variable to determine the new value for the state switch between state one and state two. Doing this for all combinations of state switches and then normalizing provides the new state transition matrix [14].

Calculating a new emission matrix simply requires the addition of 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]:

$$(Random\ HMM\ parameter\ initialization)\ \ \theta = (A, B, \pi)$$

$$(FA_1) \quad \alpha_1(i) = \pi_i * B_i(y_1)$$

$$(FA_2) \quad \alpha_i(t+1) = \left[\sum_{i=1}^{N} \alpha_j(t) * A_{ji}\right] * B_i(y_{t+1})$$

$$(BA_1) \quad \beta_i(T) = 1$$

$$(BA_2) \quad \beta_i(t) = \sum_{i=1}^{N} \beta_j(t+1) * A_{ji} * B_j(y_{t+1})$$

$$(Update\ gamma) \quad \gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$$

$$(Update\ ksi) \quad \xi_{ij}(t) = \frac{\alpha_i(t)A_{ij}\beta_j(t+1)B_j(y_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i(t)A_{ij}\beta_j(t+1)B_j(y_{t+1})}$$

$$(New\ estimation\ \pi) \quad \pi_i^* = \gamma(1)$$

$$(New\ estimation\ A) \quad A_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$$(New\ estimation\ B) \quad B_j^* = \frac{\sum_{t=1}^{T-1} 1_{y_t=v_k}\gamma_i(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

for

$$1_{y_t=v_k} = \begin{cases} 1 & if\ y_t = v_k, \\ 0 & otherwise \end{cases}$$

Where A is the state transition matrix, B is the emission matrix, π is the initial state probability vector, $y_t$ 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.

## 2.3 Optimization algorithms

### 2.3.1 Parallel Processing

A way of optimizing the computing time performance for any suitable process is by parallelizing it. This means that a global process composed of smaller individual and independent process can be sped up by passing each of the small processes to a separate central processing unit (CPU) and then combining the results at the end [40]. A diagram of how the two approaches are executed is shown in figure 2.18, where it can be seen how the parallel processing takes only 3 time steps to complete while the sequential processing needs 5. This presents a perfect theoretical case, while in reality, the parallel processing optimization is only appropriate with processes that take relatively long time. This is the case because setting the parallel processing routines of the machine can take longer than the actual computation time of the processes.

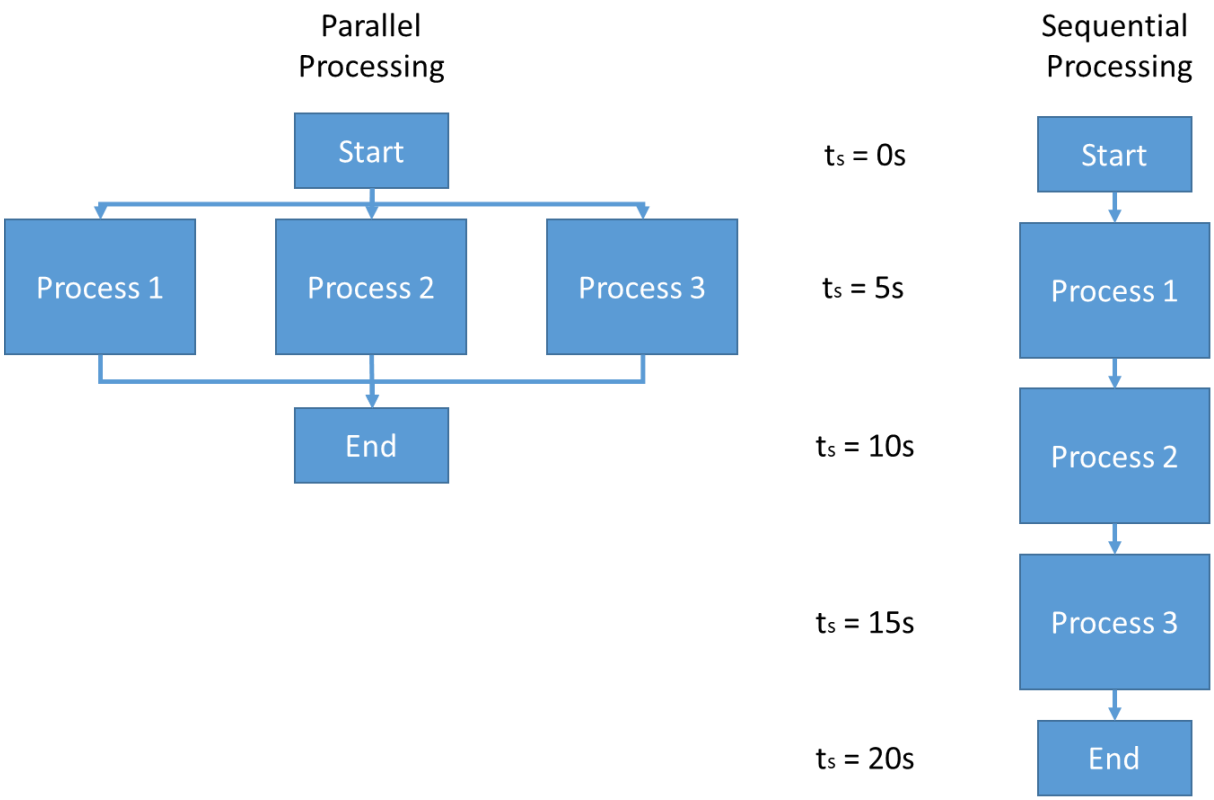



*Figure 2.18. Parallel versus sequential processing diagram. The ts indicates a time step.*

### 2.3.2 Principal Component Analysis (PCA)

PCA is a technique of dimensionality reduction, where an orthogonal transformation is applied to an n-dimensional space in order to create a new coordinate system where the first dimension has the greatest possible variance of the data points, the second the dimension the second greatest variance and so on. Figure 2.19 visualizes this transformation [31]. In the case of ML, this is extremely useful because PCA allows for portraying data points in fewer dimensions represented by the principal components while still having sufficient performance. This is due to the fact that it is often the case where some of the dimensions (features) can have a high correlation to other features or simply have a very low variance. Keeping these features increases the computational demand of the model while not providing useful information to it.

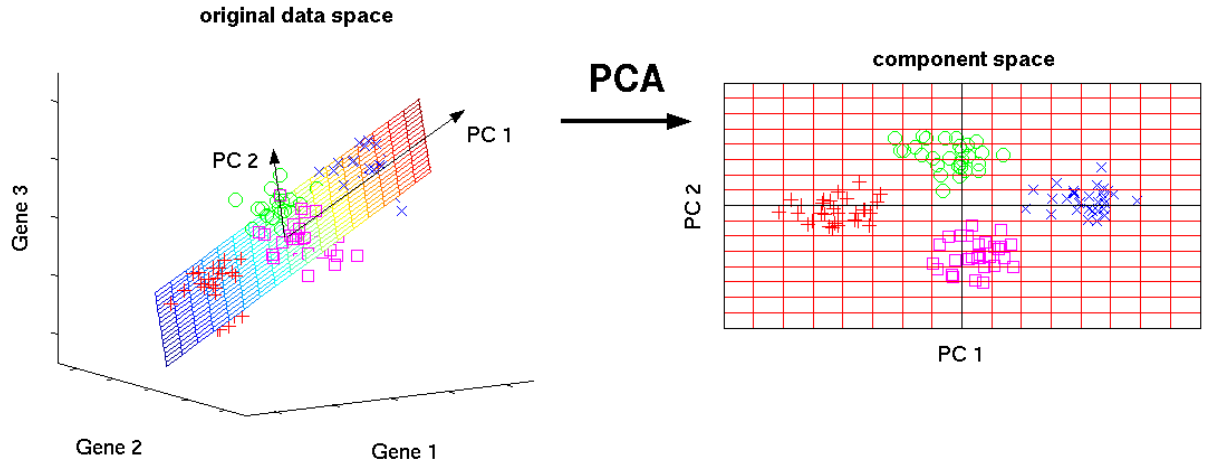



*Figure 2.19. Data in its original n-dimensional space (left) and the same data after a PCA transformation (right) with the first principal component on the x-axis and the second on the y-axis* [41]

# 3. 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, analyzing 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, the occurrence of patterns in the progression of the features correlating to the sleep stages can be expected.

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, a poor distinction between Stage 1 and REM sleep is expected, 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, it can be expected 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.

# 4. 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:

$$Sensitivity = \frac{TP}{(TP + FN)}$$

$$Specificity = \frac{TN}{(TN + FP)}$$

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

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 accuracy of the model is how many correct classifications in total were made. The specificity of the model is the proportion of actual negatives being correctly identified. The sensitivity characteristic is the true positive rate or the proportion of actual positives identified as such. This characteristic is also known as recall and probability of detection.

In addition, the confusion matrices of the models will be examined. Confusion matrices provide the TP, TN, FP, and FN in a matrix form which is indicative of how many correct classifications are made by the model in the main diagonal and also of trends in misclassification. A template of a confusion matrix can be seen in figure 27.

Another method of examining the performance is through a one-versus-all receiver operating characteristic (ROC) plot. This plot provides an idea of how accurate the classification is under varying thresholds of classification. This threshold for classification is the probability of putting a data point in the selected class. By examining each class against a combination of all the rest, we can determine how discriminative the class is and how prone for classification it is. In essence, it provides a graphical view of how sensitive and specific the model is.

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 healthy adult human 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

**Data preparation**

In order to prepare the data for training the models, it had to have the correct structure. This was done by the following procedure:

1. Read the .edf files and separate each of the channels
2. Select the appropriate EEG channel only
3. Calculate the length in data points of each epoch based on the sampling frequency
4. Pass each epoch through a band-pass filter
5. Save the epochs as columns in a .csv file
6. Repeat for all EEG channels
7. Afterward, the feature sets could be computed. This process was done as follows:
8. Read the columns (epochs) of the previously created .csv channels
9. Use a python library implementation to calculate the features based on the techniques described in section 2.2.2 and outlined below
10. Save the features for each epoch as rows in a new .csv file and append the score given by the hypnogram files.
11. Remove any rows (epochs) that have invalid values (NULL, inf, or epoch scores higher than 6)
12. Combine the feature files for all subjects into 1 large file containing all epochs for all subjects.

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 (6 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.

**Time domain features**

For the time domain feature the formulas given in section 2.2.2.1 to calculate a value for the mean, the standard deviation, the zero-crossing rate, the Hjorth mobility, and Hjorth complexity of each epoch. Additionally, the minimum and maximum value of the signal were taken as features. This resulted in a .csv file of 8 columns (one for each feature and one for the scores) and 207537 rows (one for each epoch).

**Parametric spectral features**

For the parametric spectral feature set, the first 7 AR coefficients extracted through the Yule-Walker equations provided in section 2.2.2.2 were used. This resulted in 8 columns (7 feature columns and 1 for scores) and 207537 rows (one for each epoch). The features are abbreviated as follows: AR_*, where * is the coefficient position (1-7).

**CWT spectral features**

The CWT spectral features were extracted from each band of each epoch. This resulted in 30 features (6 features for each of the 5 bands). The 6 features were the mean of the band calculated using the formula from equation 1, the variance of the spectral band given by equation 7, the Mean crossing rate, given by equation 5 with the slight change of having:

$$1_A(x) = \begin{cases} 1 & \text{if x} \in \text{ A}, \\ \mu & \text{if } x \in A. \end{cases}$$

the total power of the band given by the integrating the signal, the spectral edge frequency giving the frequency below which 95% of the power spectrum falls, and the relative spectral power which is given by the power of the band divided by the absolute power of the whole spectrogram. These features were abbreviated as follows:

- spectral mean: *m
- spectral variance: *v
- spectral mean-crossing rate: *mcr
- total spectral power: *p
- relative spectral power: *r
- spectral edge frequency: *f7

where * stands for the EEG band (D for Delta, T for Theta, A for Alpha, B for Beta and G for Gamma)

**Training the models**

After the features have been calculated and saved as a .csv they were fed into the ML algorithms described in section 2.2.3. This was done in the following manner:

1. Read the .csv file containing all epochs represented by their feature set and their score.
2. Separate the data into features and scores
3. Separate the data into training and testing sets with a ratio of 79:21
4. Feed the training data into an ML algorithm implementation (RF taken as a base reference) to train a model
5. Calculate the Accuracy, Sensitivity, and Specificity of the model
6. Do a cross-validation check (10 fold)
7. Repeat for all channels of the data.

### 4.2.2 Evaluation of different ML algorithms

In this experiment, the goal was to examine which of the ML algorithms provides the highest performance results. Therefore, the same procedure as in the previous experiment was followed for the 4 separate types of models: 3 class RF, 3 class KNN, 6 class RF, 6 class KNN.

### 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 from 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 Combined feature set evaluation

After all independent feature sets were analyzed, they were combined as to form one large .csv file containing 45 columns (7 time domain features, 7 parametric spectral features, 30 CWT spectral features,1 epoch scores) and 207537 rows (one for each epoch). Then the same procedure as in the previous tests was followed.

### 4.2.5 Optimization

**Parallel processing**

The parallel processing experiment was simply recording the computational times of converting the .edf files into .csv files. The parallelization of the process was rooted in treating each of the channels as an independent process because they have independent data. Therefore, the parallel process involves all channels being converted simultaneously while the sequential process treats them one by one. The number of channels is 10 and the number of parallel processes available on the machine is 8 so 2 processes had to be treated sequentially anyway.

**PCA**

For the PCA experiment, the combined feature set was used and fed through a PCA implementation in the sklearn library form Python. After the principal components were calculated, a varying number of them was fed as a feature set to all 4 types of models. Afterward, the results are analysed as in the previous tests.

### 4.2.6 Hidden Markov Model evaluation

The HMM experiment was performed in the following way:

1. Read the combined features .csv file
2. Use the all 44 features as an observation in a Gaussian Hidden Markov Model
3. Feed the resulting observation sequence into the Baum-Welch EM algorithm described in section 2.2.3 in order to estimate the HMM
4. Use the Viterbi algorithm on the trained HMM to predict the most likely sequence of hidden states

After the HMM has yielded its results the same analysis as in the previous tests was performed.

### 4.2.7 Robustness check with the DREAMS dataset

So far all tests have been done on the Hut lab dataset. The robustness test (also to be referred to as cross-dataset model), uses both the Hut lab dataset and the DREAMS dataset. In this case, the DREAMS dataset is treated in the same way as the Hut dataset, converting the .edf files to .csv files, extracting the individual features and combining them into one large combined features dataset of 45 columns and 121618 epochs. Afterward, the training data is chosen to be all the epochs from the Hut dataset and the testing data all of the epochs of the DREAMS dataset. The same analysis as before is performed on the results

# 5. RESEARCH RESULTS

## 5.1 Hut lab dataset characteristics

The first step in the process of training an ML model is understanding the data. The ’10 second Hut lab’ dataset consists of the full night sleep EEG recordings of 50 healthy human adults. The signals are collected through 5 electrodes: Oz, Cz, Fpz, C3, and C4 which are outlined in figure 2.1. These 5 electrodes form 10 separate EEG channels: Oz-Cz, Oz-Fpz, Oz-C3, Oz-C4, Cz-Fpz, Cz-C3, Cz-C4, Fpz-C3, Fpz-C4, C3-C4.

The signals in these channels are different but the score is assigned to an epoch and therefore will be the same for the respective epochs throughout all channels. The total number of 10-second epochs for all subjects is 209487. The distribution of epochs per sleep stage is shown in figure 5.1. It can clearly be seen how they are not evenly distributed, with the ‘Wake’, ‘S2’, and ‘REM’ stage being represented by a relatively higher number of epochs. Additionally, it can be seen that there is also an ‘unknown’ label. This label exists to classify noisy epochs, where the EEG signal is corrupted. These corruptions are called artifacts and need to be discarded because they skew the features which are extracted from the signals. After the removal of the corrupted epochs, their total number is reduced to 207537.

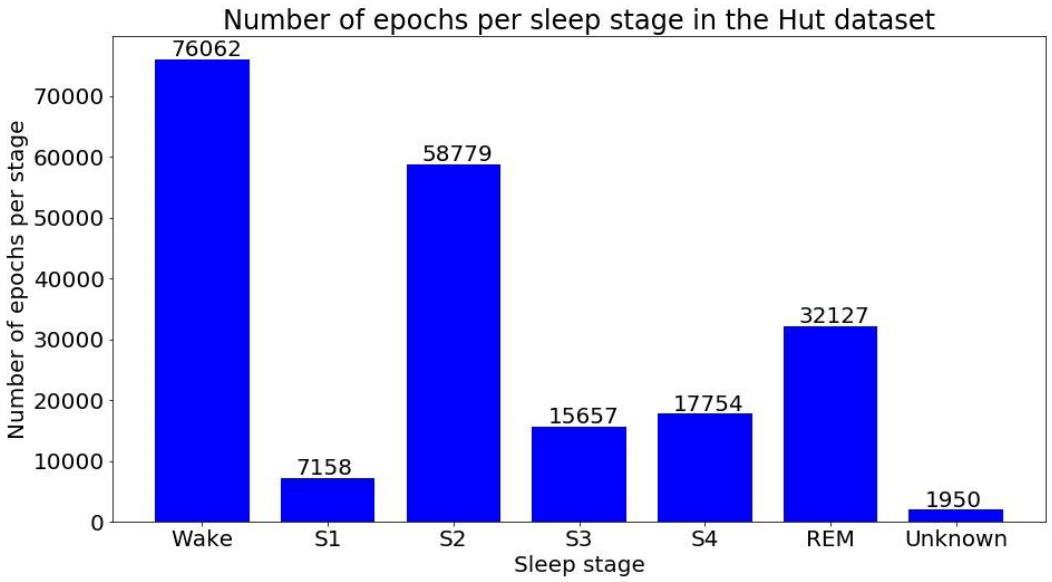



*Figure 5.1. Number of 10-second epochs per sleep stage for the full dataset of 50 subjects.*

It is also important to see how the sleep stages evolve during the night in the subjects from the database and how well that compares to the typical model that can be seen in figure 2.4. Unfortunately, the average of all the subjects cannot be taken because their sleep is largely varying. However, several individual subjects can be selected and their hypnograms examined, as shown in figure 5.2. No patterns are immediately apparent by looking at the figure. The hypnograms for all subjects can be found in Appendix A.

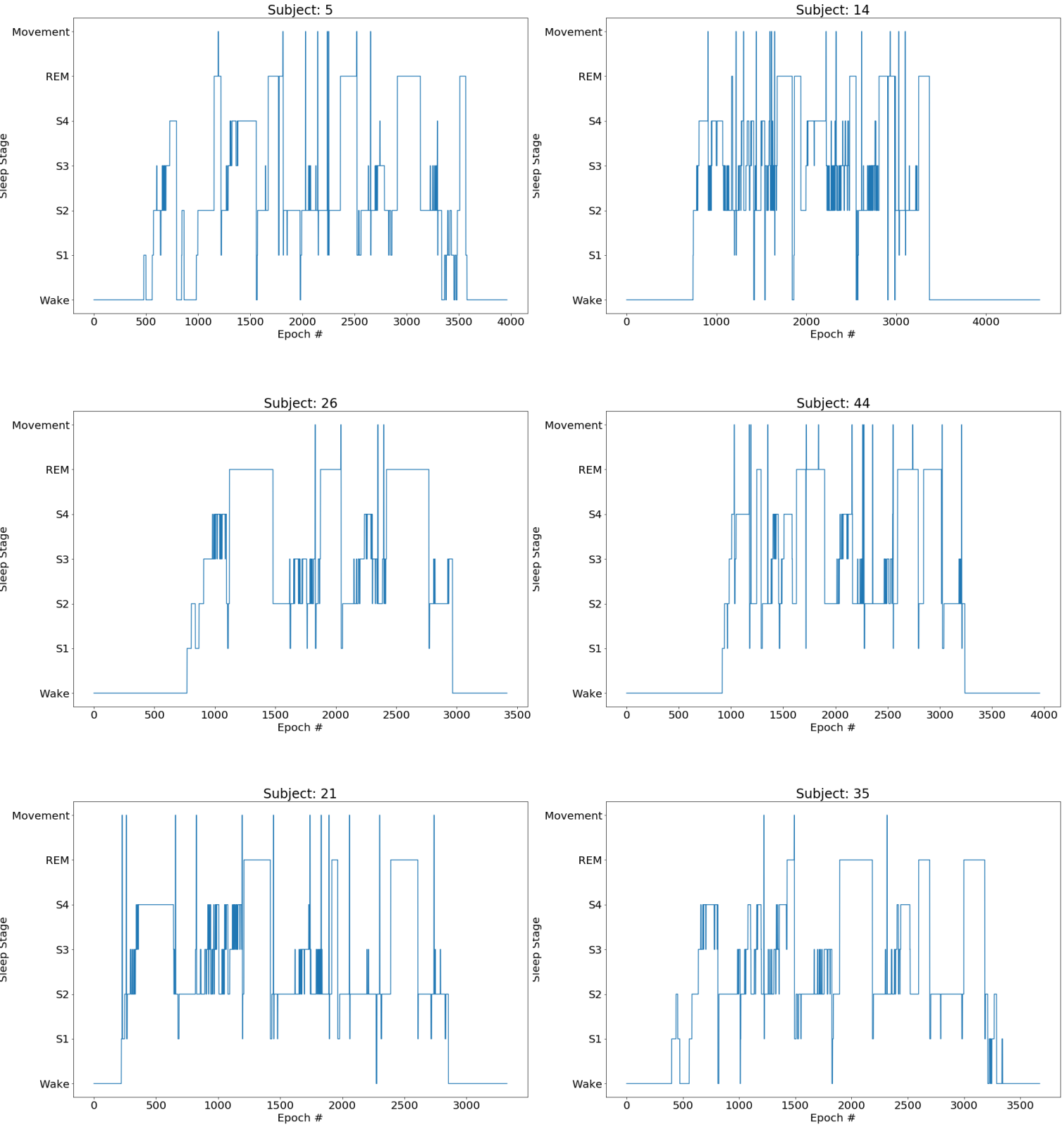



*Figure 5.2. Hypnograms of randomly selected subjects.*

An additional step was required before the features could be extracted. The raw EEG signals needed to be filtered with a bandpass filter to remove the frequencies that were both too high and too low. The filter used was a 5$^{th}$ order bandpass Butterworth filter. The response of the filter is shown in figure 5.3.

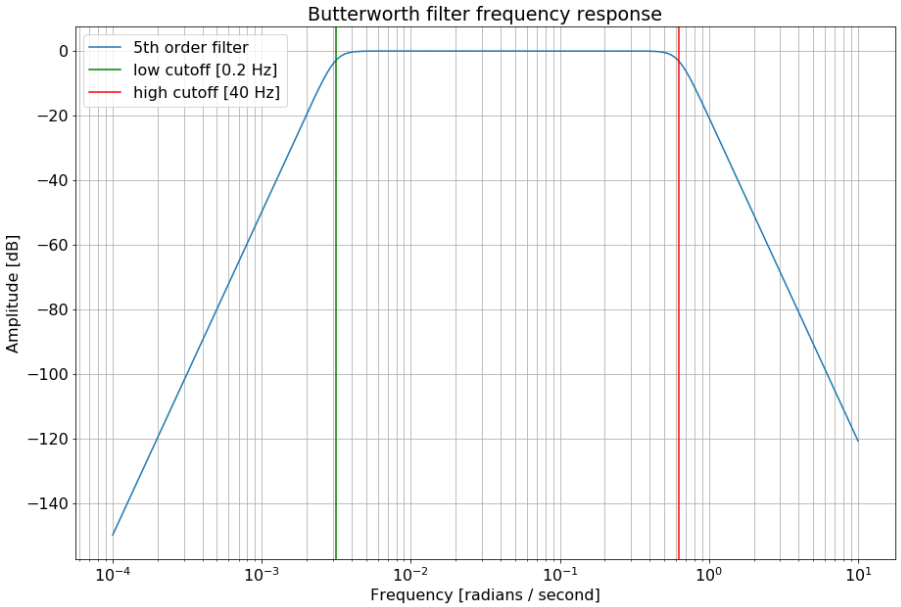



*Figure 5.3. 5th order Butterworth bandpass filter frequency response used for filtering the raw EEG signals before feature extraction.*

Filtering the signals is required because it eliminated some of the noise which might influence the extracted features and consequently the overall performance of the ML algorithms and their models. That being said, the effects of the filter are not really influential as most of the filtered data is from the high frequencies. Figure 5.4 shows a typical raw EEG signal and the same signal after filtering. It can be seen that there is a phase shift in the filtered signal but that should not be influencing the features.

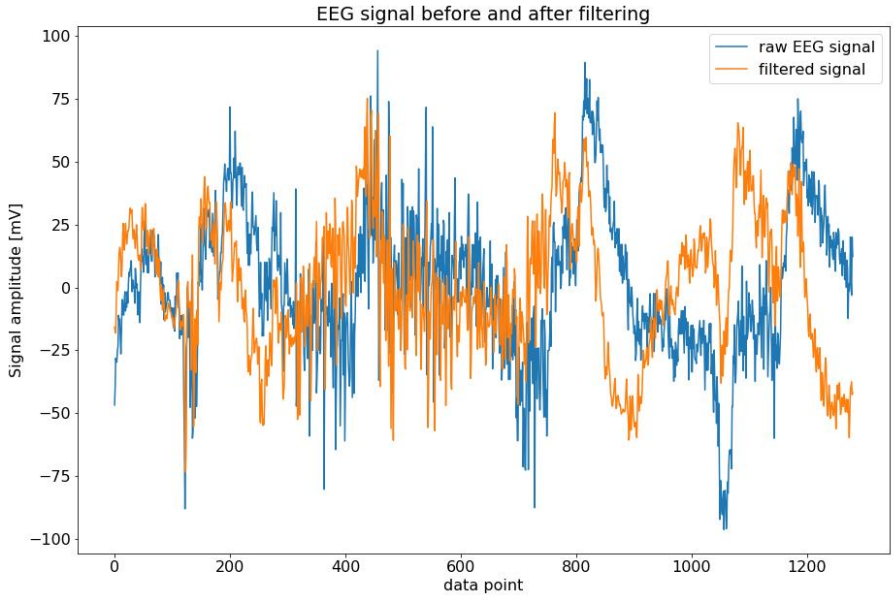



*Figure 5.4. Raw EEG signal (blue) and the same signal after the 5th order bandpass Butterworth filter from figure 5.3.*

## 5.2 Feature set evaluation and channel evaluation

### 5.2.1 Time domain features

As 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 5.5 illustrates the average value of each of these features for the different classes for the classifier algorithm. This gives an initial idea of how well these features would segregate the classes. It can be seen 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.

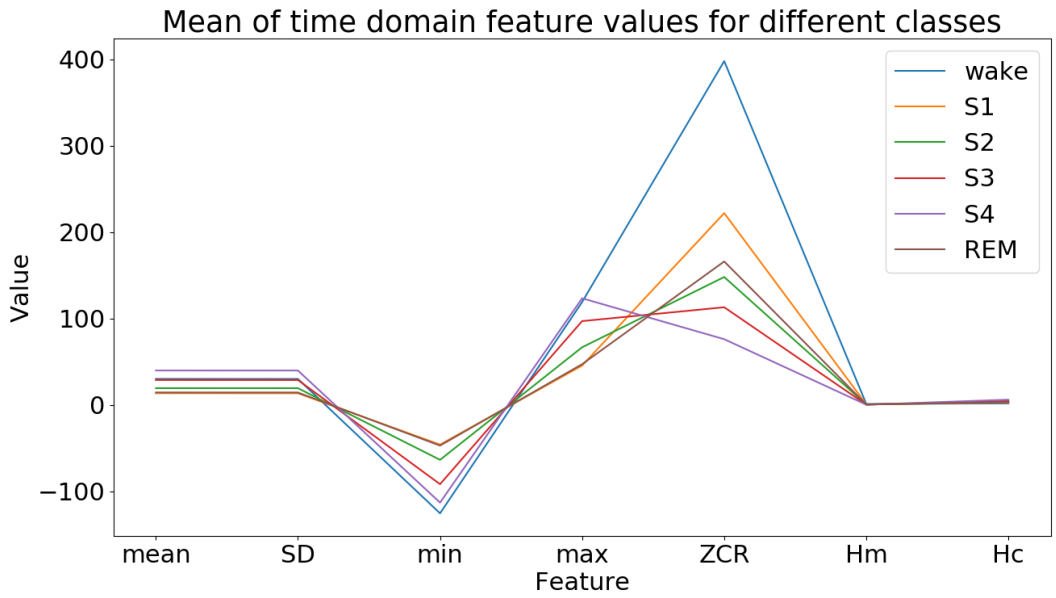



*Figure 5.5. Feature variation across different classes. The features with the highest variance in mean values are best suited for segregation of the classes.*

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 5.6 is a better representation of how well the feature sets will segregate classes. As with figure 5.5 here it can also be seen 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.

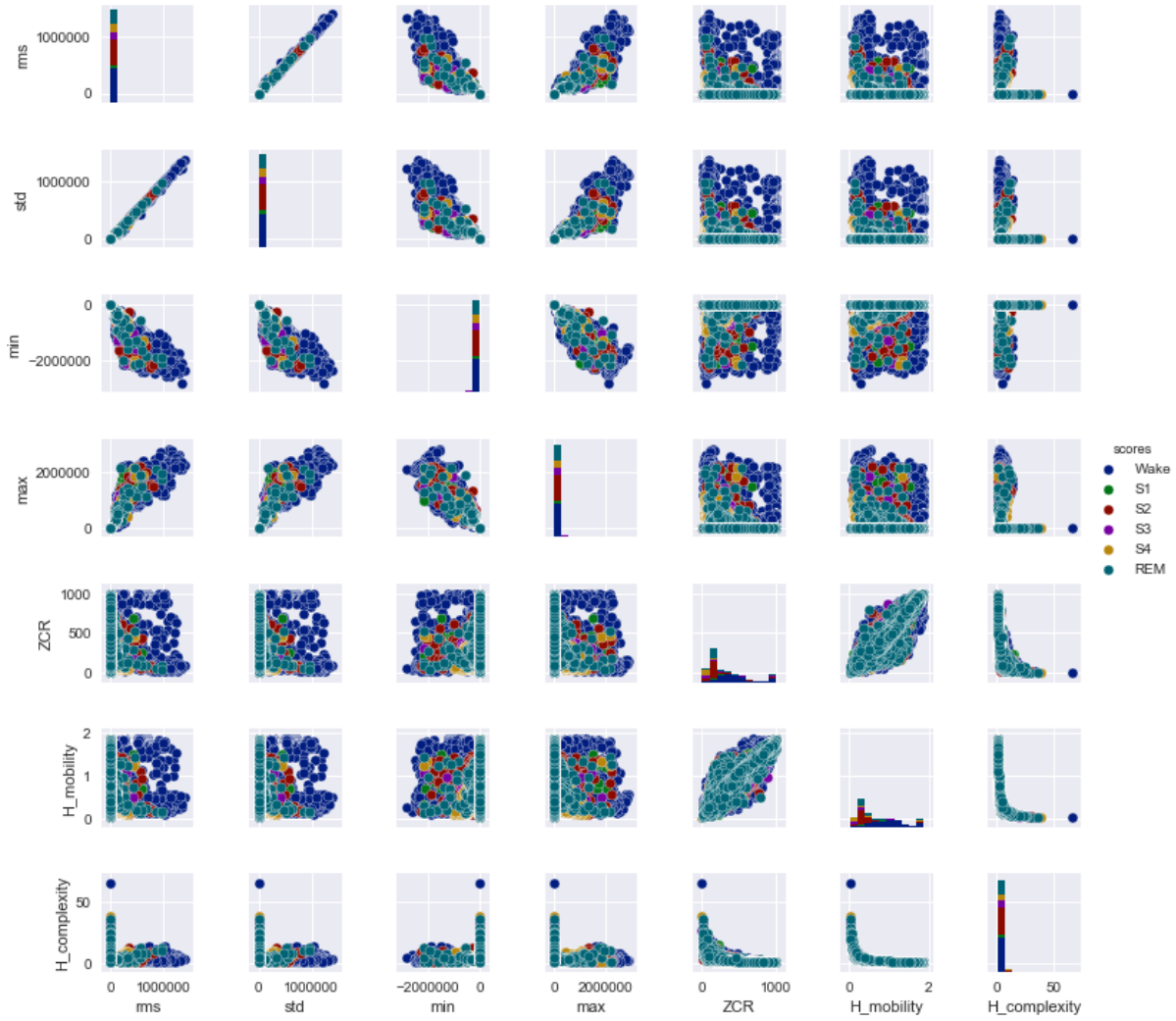



*Figure 5.6. Pair plot of the time domain features. The plot shows the relationship between separate feature sets. Ideally, the data points with the same colours would form separate clusters, therefore larger variance is desirable.*

Now a better idea of the characteristics of the feature set exist, a model can be trained to classify the sleep stages. In this case, an RF and a 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 79:21. 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. It is seen from the results in figure 5.7 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, it can be seen 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 it can clearly be seen that all three electrodes (Cz, C3, and C4) are on top of the head, as their abbreviations suggest: C for Centre.

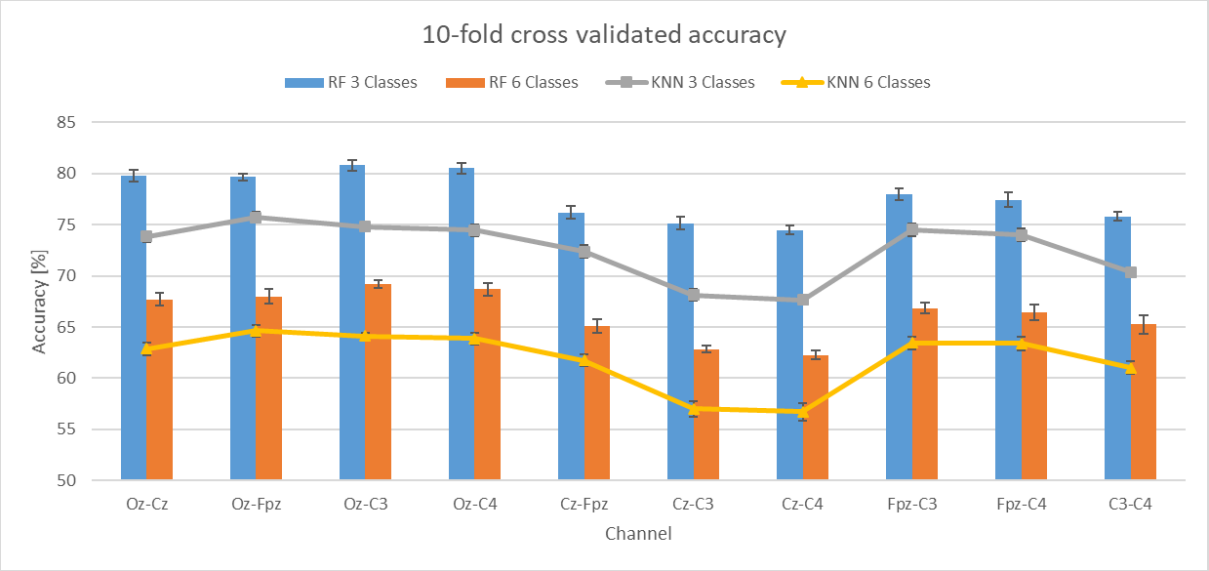



*Figure 5.7. 10-fold cross-validated Accuracy with error ranges for each EEG channel of the 10-second Hut lab data for the time domain feature set under both a Random Forest and a K-nearest neighbour (k=30) classifiers for both 3 and 6 classes.*

While this is an impressive result, given the feature set fed into the algorithm it is important to understand where the errors come from. The confusion matrices shown in figure 5.8 helps analyze this. It can be seen from the first matrix that the Wake class has a very high rate of true positives. The S2 and S4 classes are also relatively 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 only 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 it can be seen from 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.

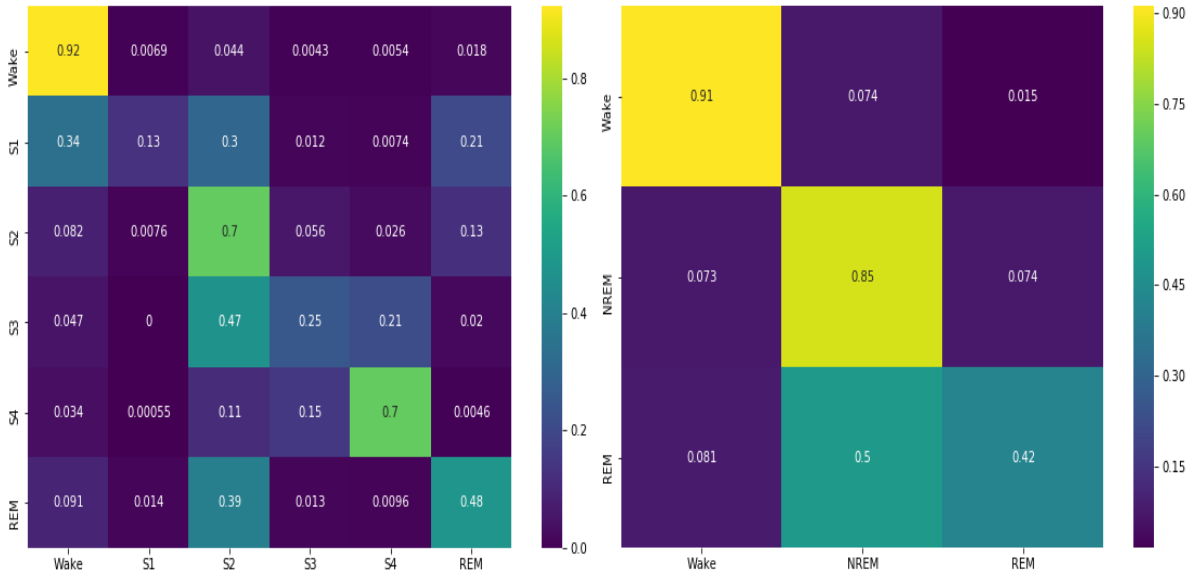



*Figure 5.8. Normalized confusion matrix for 6 class (left) and 3 class (right) RF classifiers of the Oz-C3 channel for the time-domain 10-second Hut lab data model.*

Before moving onto the next type of features a 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 B as well as all other Sensitivity and Specificity tables. 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. It can be seen 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 that the models are too specific. This means that almost all of the negatives are correctly labeled as negatives. However, it also means that many positives are also labeled as negatives.

*Table 5.1. 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.*

| Channel | | Sensitivity | | | | | | Specificity | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Wake | NREM | | | | REM | Wake | NREM | | | | REM |
| | | | S1 | S2 | S3 | S4 | | | S1 | S2 | S3 | S4 | |
| Oz-C3 | RF 3 class | 0.9 | | 0.9 | | | 0.4 | 0.9 | | 0.8 | | | 1 |
| | RF 6 class | 0.9 | 0.1 | 0.7 | 0.3 | 0.7 | 0.5 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |
| | KNN 3 class | 0.9 | | 0.8 | | | 0.3 | 0.9 | | 0.8 | | | 1 |
| | KNN 6 class | 0.9 | 0.1 | 0.7 | 0.1 | 0.7 | 0.4 | 0.8 | 1 | 0.8 | 1 | 1 | 0.9 |
| Oz-Fpz | RF 3 class | 0.9 | | 0.9 | | | 0.4 | 0.9 | | 0.8 | | | 1 |
| | RF 6 class | 0.9 | 0.1 | 0.7 | 0.2 | 0.7 | 0.4 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |
| | KNN 3 class | 0.9 | | 0.8 | | | 0.2 | 0.9 | | 0.7 | | | 1 |
| | KNN 6 class | 0.9 | 0.1 | 0.7 | 0.1 | 0.7 | 0.3 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |
| Cz-C4 | RF 3 class | 0.8 | | 0.8 | | | 0.3 | 0.9 | | 0.7 | | | 0.9 |
| | RF 6 class | 0.9 | 0.1 | 0.6 | 0.2 | 0.6 | 0.4 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |
| | KNN 3 class | 0.8 | | 0.8 | | | 0.2 | 0.8 | | 0.7 | | | 1 |
| | KNN 6 class | 0.8 | 0 | 0.6 | 0 | 0.6 | 0.3 | 0.7 | 1 | 0.8 | 1 | 1 | 0.9 |

Looking at the data can give 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 this 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 to 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 5.9. 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.

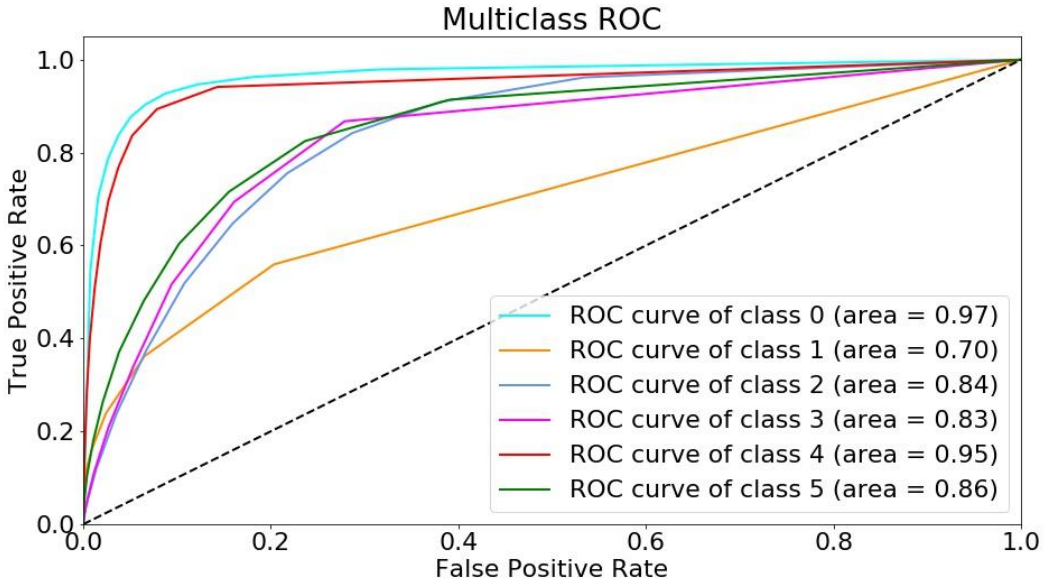



*Figure 5.9. ROC curves for all classes under the Oz-C3 channel for 6 classes using RF. The classes go as follows: 0-Wake ;1-S1;2-S2; 3-S3; 4-S4; 5-REM*

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 5.10. It can be seen that the ZCR, Hjorth mobility and Hjorth complexity have a higher overall importance than the rest.

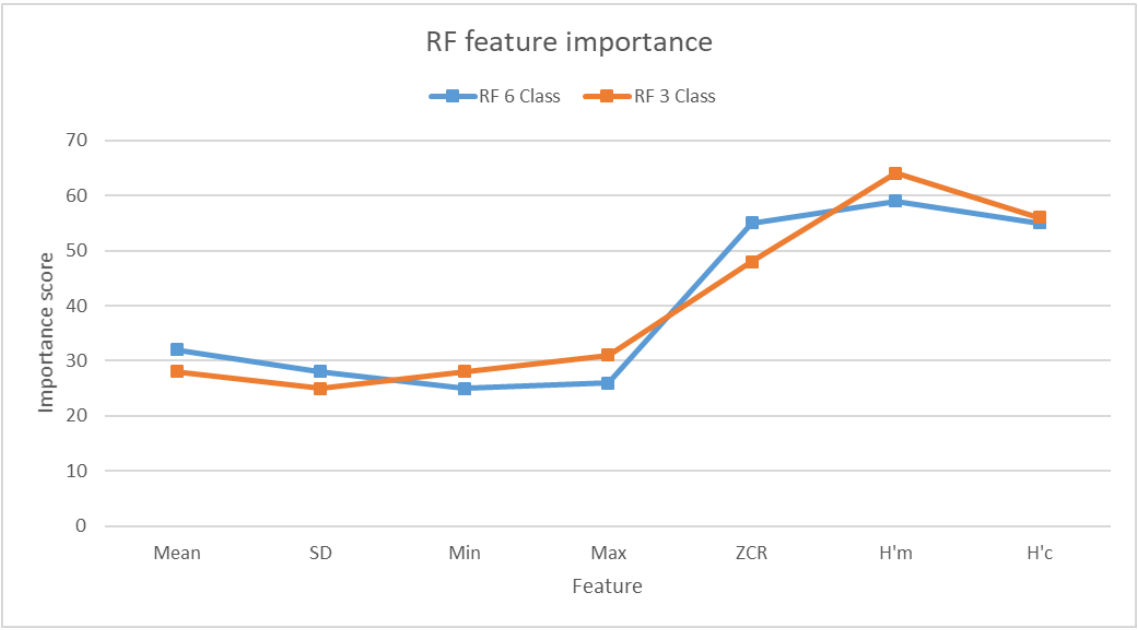



*Figure 5.10. Feature importance for the RF algorithm derived through summing the importance of the features across all channels.*

### 5.2.2 Parametric features

As with the features from the time domain, a first look at the distribution of the features is given in figure 5.11

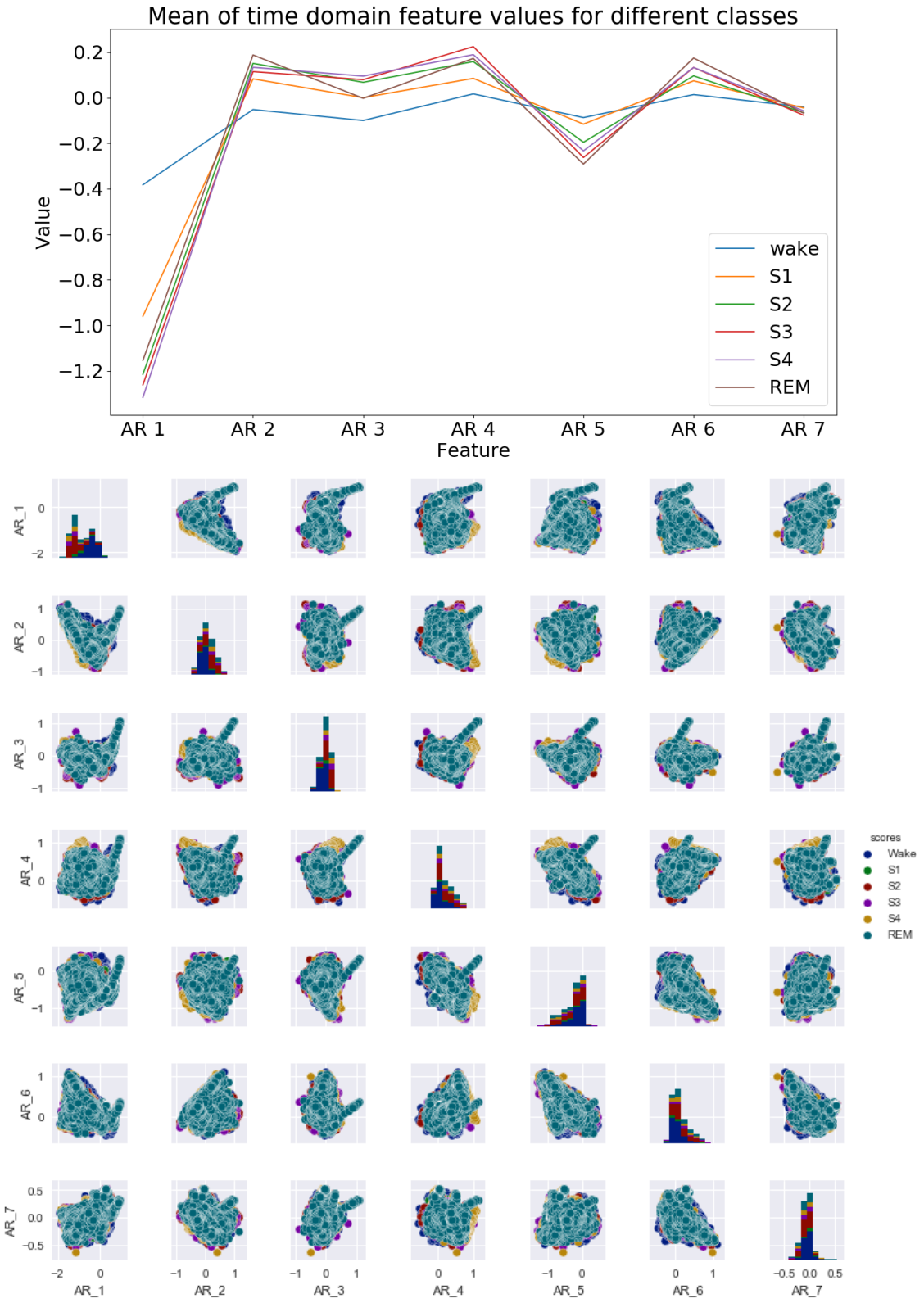



*Figure 5.11. Means of the features for the different classes (top), and the respective pair plot showing the distribution of the features in respect to each other.*

These distributions suggest that the first defining weight will have the highest importance because of the largest distribution that it presents. Looking at the pair plot it can be seen that the features are largely overlapping and creating a single cluster. Looking at the distributions on the diagonal, however, presents a larger variance in the classes than what is seen in the time domain features, where only the ZCR and the Hjorth mobility had some spread instead of a single column. Moreover, it can be seen from the means of the coefficients that their variance starts to decrease with further iterations into the latter coefficients.

For training the models, the same conditions as in the previous case apply. The k value for the KNN is 30 and both algorithms (RF and KNN) are trained both for 6 and 3 classes. 10-fold cross validation is done to ensure that the performance is not yielded by an extreme case. The results are shown in figure 5.12. It can be seen that the performance is highest under the Oz-C4 channel in the 3 class KNN model and lowest for the Cz-C3 channel in the 6 class RF model. The things to note in this figure are how the performance of the KNN is greater in comparison to the RF in this case under all conditions in contrast to the time-domain features. However, it must also be said that the channels yielding the highest accuracies remain Oz-C3 and Oz-C4 with the Cz-C3 being the worst. An interesting detail is the drop in accuracy under the Oz-Fpz channel which had the highest KNN performance in the time domain features. Other than the dip in the Oz-Fpz, it seems that the Oz channel provides the best conditions for extracting parametric features.

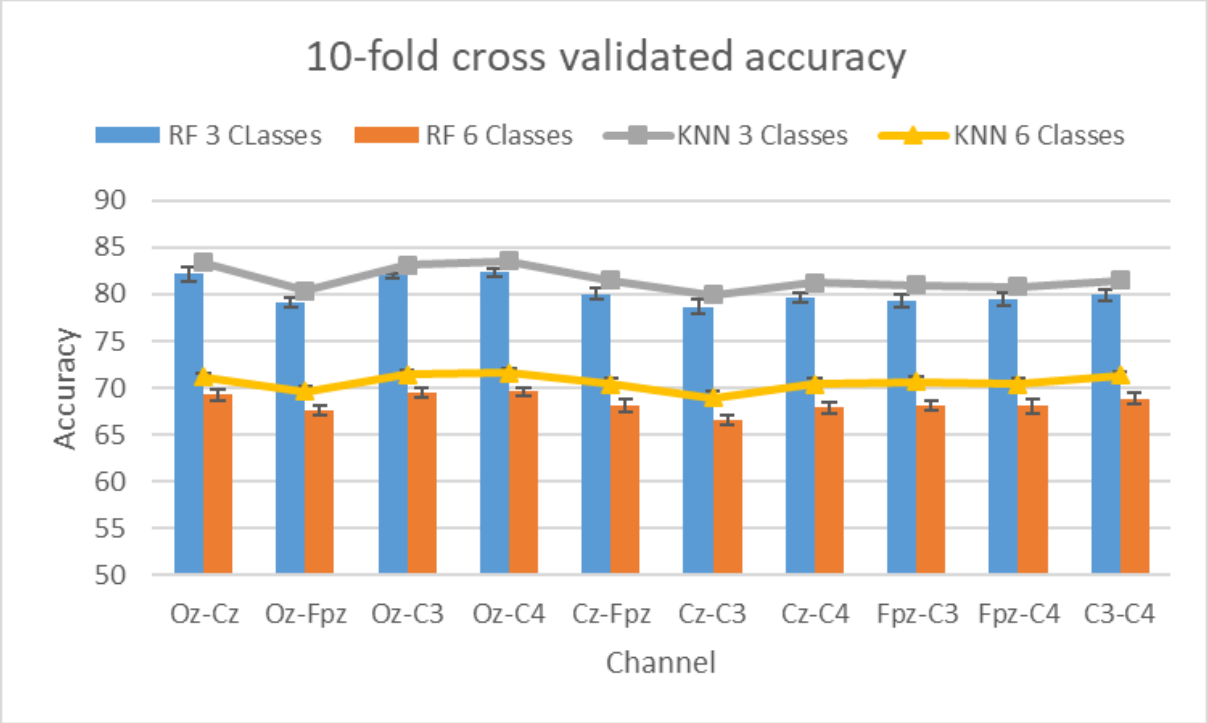



*Figure 5.12. 10-fold cross-validated Accuracy with error ranges for each EEG channel of the 10-second Hut lab data for the parametric frequency domain feature set under both a Random Forest and a K-nearest neighbor (k=30) classifiers for both 3 and 6 classes.*

The confusion matrices of the Oz-C4 channel under the KNN are given in figure 5.13. The values in them are largely similar to the ones from the time domain RF Oz-C3 matrix shown in figure 32. The notable difference can be seen in the increase of misclassified S3 and S4 epochs and the increase of correctly classified REM epochs. In this case, there is a drop of correctly classified S3 and S4 epochs which seem to be often classified as S2. The slight increase in S2 and the distribution of misclassifications in these three epochs for S3 and S4 supports that. The increase of correctly

classified REM stages is by 16%. Additionally, there is a drop of REM epochs classified as S2 by 11%. As for the 3 class matrix, the only notable change is the increase of correctly classified REM epochs. The increase is by 16% as with the 6 classes and a drop of NREM misclassification of 15% in total.

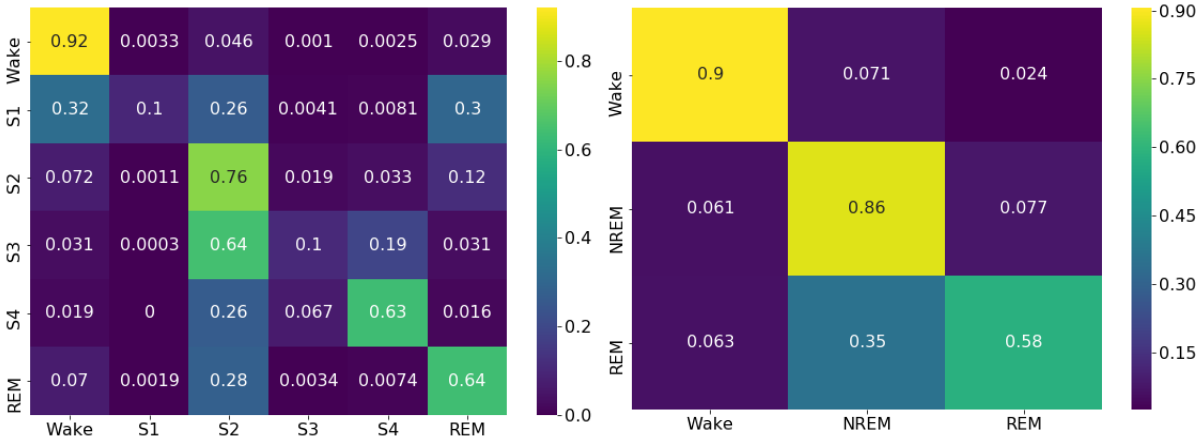



*Figure 5.13. Normalized confusion matrix for 6 class (left) and 3 class (right) KNN classifiers of the Oz-C4 channel for the parametric frequency domain 10-second Hut lab data model.*

The Sensitivity and Specificity of the best and worst channels can be seen in table 5.2. As before it can be seen that there is a high Specificity in all cases. As with the time domain, the best results are shown in green while the changes for the worst are outlined in red. Again it can be seen that dropping the sensitivity results in a worse performance. This makes sense since the models become less sensitive but remain highly specific. This means that while the model accepts fewer data points as positive it keeps having the same threshold for negative classification. There is also a drop in sensitivity as compared to the time features in the S3 and S4 classes and an even more notable increase in the sensitivity for the REM class.

*Table 5.2 Sensitivity 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.*

| Channel | | Sensitivity | | | | | | Specificity | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Wake | NREM | | | | REM | Wake | NREM | | | | REM |
| | | | S1 | S2 | S3 | S4 | | | S1 | S2 | S3 | S4 | |
| Oz-C4 | RF 3 class | 0.9 | | 0.9 | | | 0.5 | 0.9 | | 0.8 | | | 1 |
| | RF 6 class | 0.9 | 0.1 | 0.7 | 0.2 | 0.6 | 0.6 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |
| | KNN 3 class | 0.9 | | 0.9 | | | 0.6 | 0.9 | | 0.8 | | | 0.9 |
| | KNN 6 class | 0.9 | 0.1 | 0.8 | 0.1 | 0.6 | 0.6 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |
| Cz-C3 | RF 3 class | 0.8 | | 0.8 | | | 0.5 | 0.9 | | 0.8 | | | 0.9 |
| | RF 6 class | 0.9 | 0.1 | 0.7 | 0.2 | 0.6 | 0.5 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |
| | KNN 3 class | 0.8 | | 0.9 | | | 0.5 | 0.9 | | 0.8 | | | 0.9 |
| | KNN 6 class | 0.8 | 0.1 | 0.8 | 0.1 | 0.7 | 0.6 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |

The ROC plot seen in figure 5.14, supports the results seen in the confusion matrices and the table. It can be seen that the S1 class has the worst performance with the Wake having the highest again. However, there is a notable difference in the areas under the curves. The area remains the same for the Wake and the S2 classes. There is an increase in the area under the REM class and the S1 class. The area under S3 and S4 has dropped. This confirms the hypothesis that there is an increase in the epochs which are classified as REM in general even if the specificity remains the same. The fact that the REM stage classification has improved under unchanging algorithm parameters means that the features are better suited to represent the REM sleep stage. They are successfully pulling the REM epochs above the threshold for classification, therefore, yielding higher specificity and lower false positive rate.

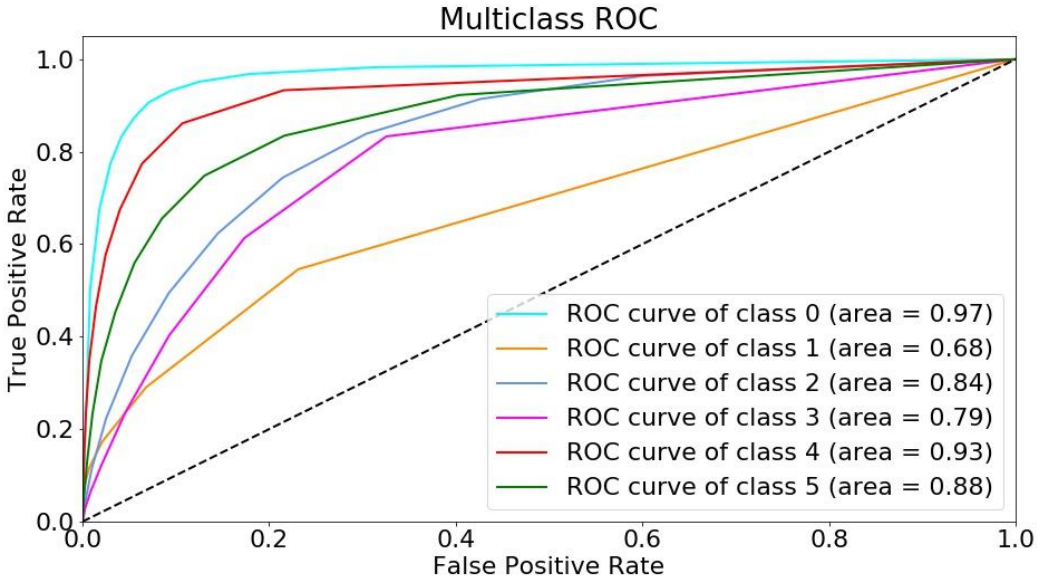



*Figure 5.14. ROC curves for all classes under the Oz-C3 channel for 6 classes using RF. The classes go as follows: 0-Wake ;1-S1;2-S2; 3-S3; 4-S4; 5-REM*

The final evaluation of the parametric features is by their importance. The same method as before is used where a score is given to the feature based on the number of times the feature occurs at the respective positions for all channels. The results are given in figure 5.15. It is clear that the first coefficient is the most dominant with the next 3 having some impact as well.

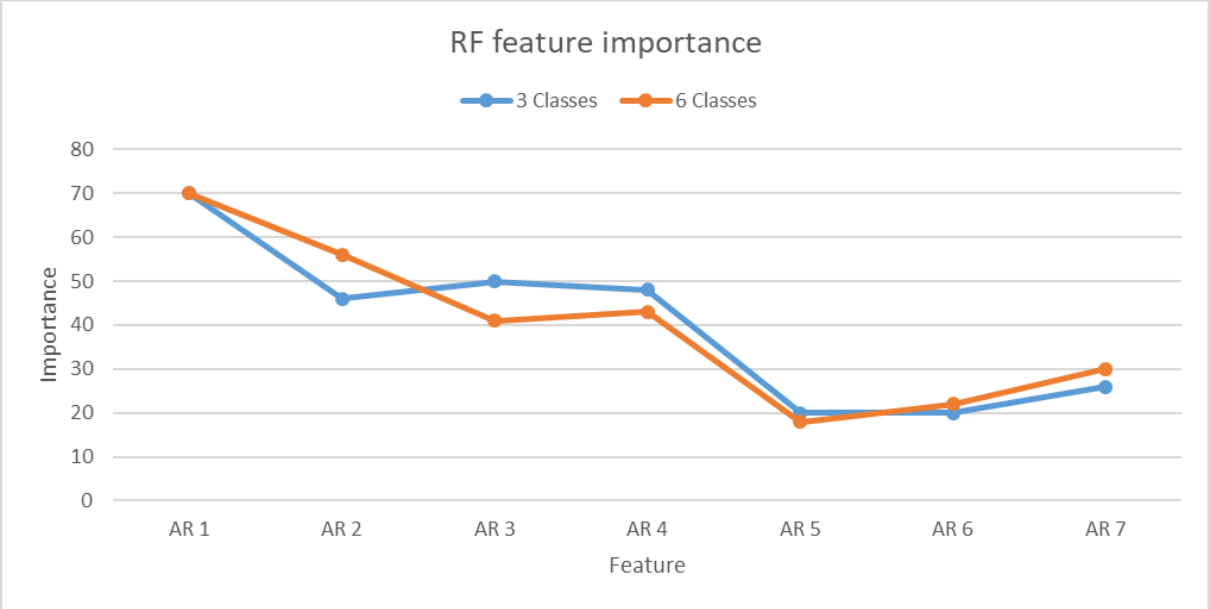



*Figure 5.15. Feature importance for the RF algorithm derived through summing the importance of the features across all channels.*

### 5.2.3 Continuous wavelet transform features

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. Therefore, each of the sub-figures in figure 5.16 has the same feature but for all frequency bands. The features go as follows: Spectral mean, Mean-crossing rate, Total band power, spectral inequality, spectral variance, and spectral edge frequency. The bands go as follows: Delta, Theta, Alpha, Beta, Gamma, representing each row/column in each sub-figure.

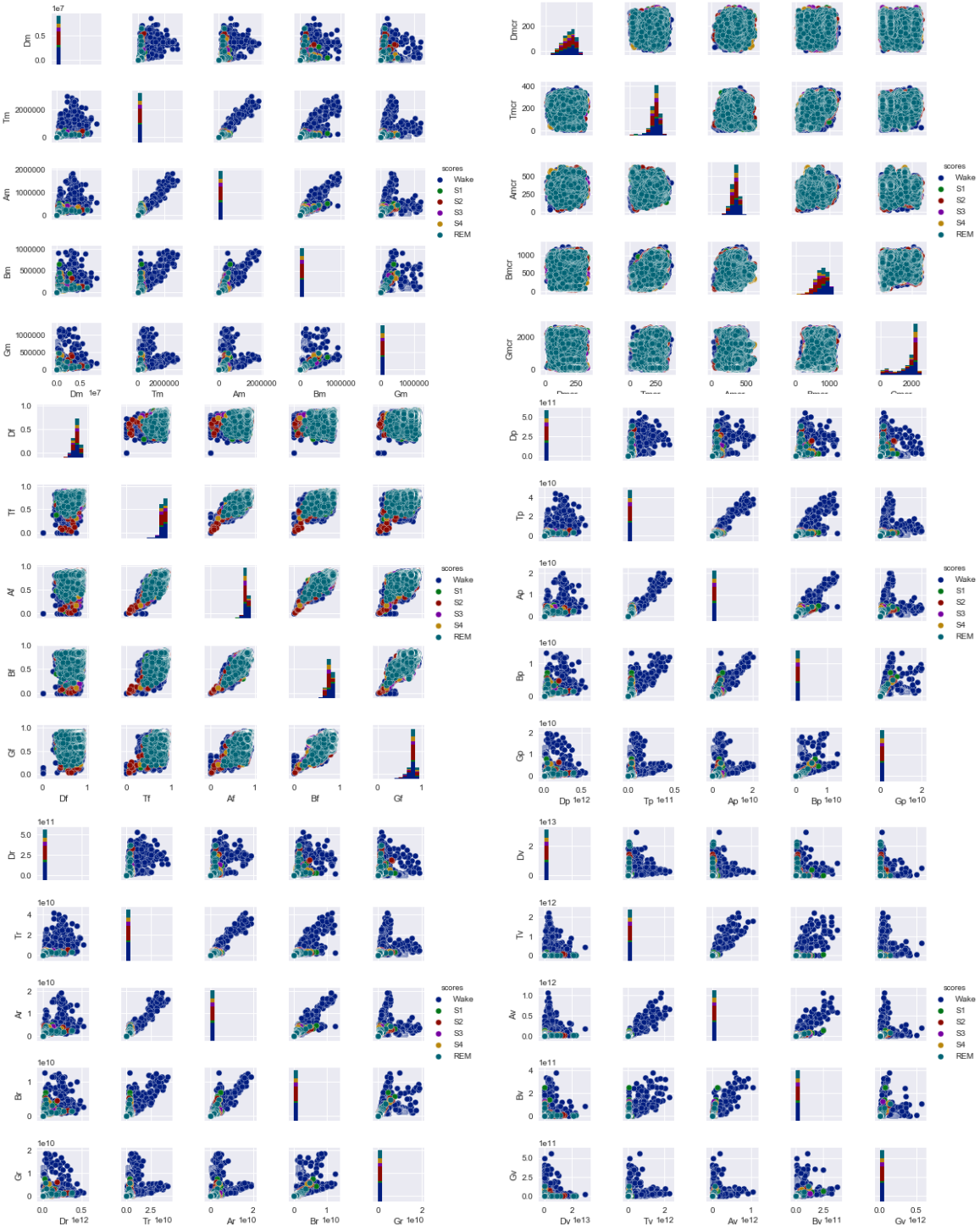



*Figure 5.16. distributions of different features across the bands: means (top left), middle-crossing rate (top right), total band power (middle left), relative spectral power (middle right), variance (bottom left), spectral edge frequency (bottom right)*

In this case, it can be seen some separation in the colours in the pair plots. The total band power and the Mean crossing rate provide the widest distributions. It can be expected that the total band power will have the highest importance of all features. It is unfortunate that displaying all features against each other to see how the distributions will overlap is impossible but this feature set contains 30 features instead of the small feature sets presented so far.

The performance of the different models can be seen in figure 5.17. It is clear that the overall performance has increased almost reaching 90%. The conditions for testing remain the same as in the previous two tests. A notable difference is the drop in performance in the Cz-Fpz channel for the KNN models. Interestingly, the lowest performance for the RF models is not in the same channel or even associated with the same areas of the brain. The lowest accuracy for the RF models is the Cz-C4 as with the previous tests. The highest value is given by the Fpz-C3 channel on the RF 3 class model. Here, again there is a disagreement between the ML algorithms, with the best performance for the KNN models being under the Oz-C4 channel. Again as with the parametric features, it can be seen a drop in performance in the Oz-Fpz channel in both KNN models. Overall, the Cz channels have the worst performance with the best performance given by the ones involving C3 and C4.

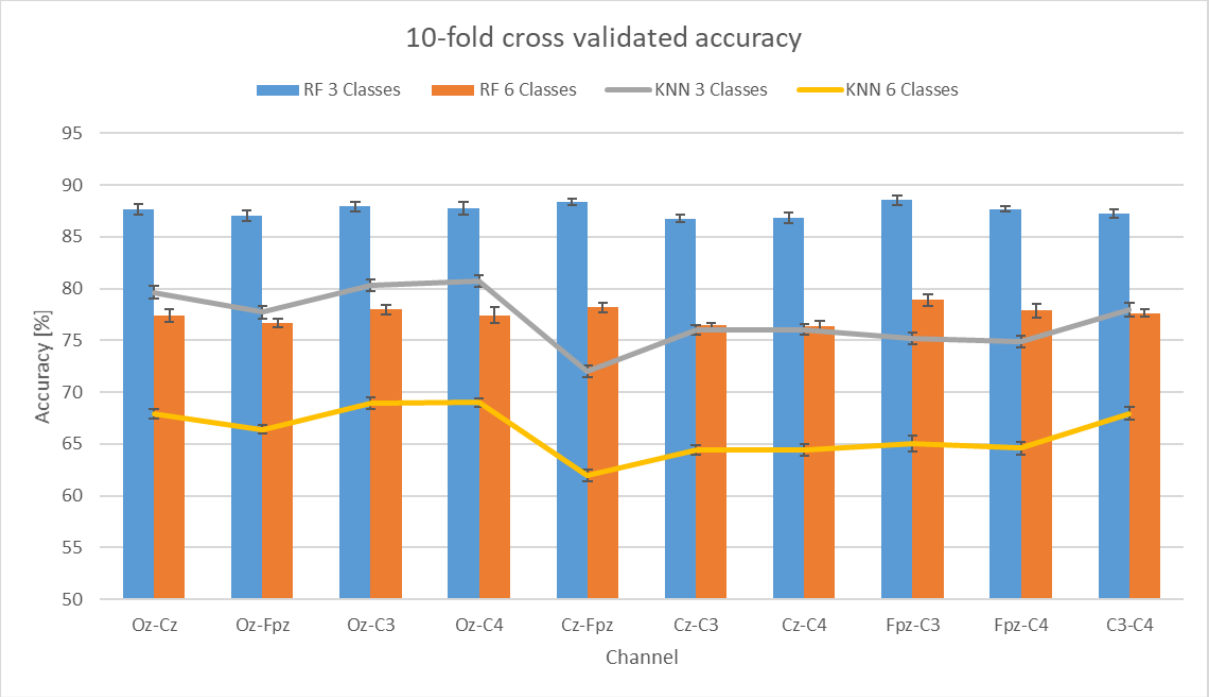



*Figure 5.17. 10-fold cross-validated Accuracy with error ranges for each EEG channel of the 10-second Hut lab data for the CWT time-frequency domain feature set under both a Random Forest and a K-nearest neighbour (k=30) classifiers for both 3 and 6 classes*

The confusion matrices for the Fpz-C3 channel are provided in figure 5.18. There is a slight increase in the Wake recognition by 3% compared to both of the previous methods. The same issues as the previous two sets of features for the S1 class are present here as well. There is an increase in the classification of the S2 class. There is a relative drop in the S3 confusion with S2 in this case and a slight increase in the correct classification of the S3 class. Still, more S3 epochs are classified as S2 in this case as well. There is an increase in the correct classification of the S4 class, with most of the confusion happening with the S3 class. This is expected and practically a positive indicator for the model. There is also a notable increase in the correct classification of the REM stages. As for the 3 class matrix, there is an overall improvement in all cases with the only confusion coming from the incorrectly classified Rem stages. This happens because 19% of the REM stages are classified as S2.

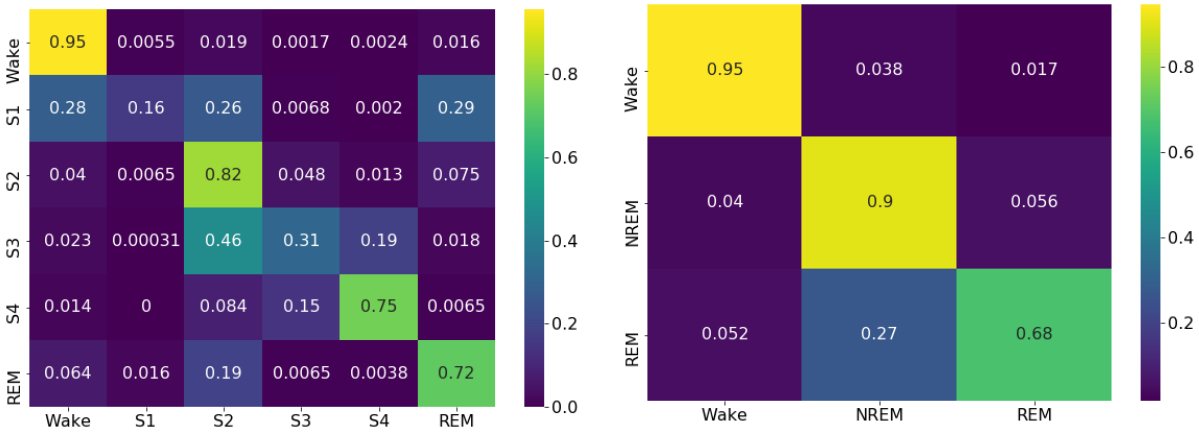



*Figure 5.18. Normalized confusion matrix for 6 class (left) and 3 class (right) KNN classifiers of the Fpz-C3 channel for the parametric frequency domain 10-second Hut lab data model.*

Table 5.3 provides the Sensitivity and Specificity values of the best channels, shown in green, and the changes in the worst channels, shown in red. Similarly, to the previous tests, the sensitivity drops in the worst cases as expected. The values for the Sensitivity have also increased in comparison with the previous tests, across all stages. As before, the Specificity remains high, which can be explained by the same reasoning. The fact that the problem is not binary, means that the classification threshold is set too high in the algorithms and the specificity is therefore high.

*Table 5.3. 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 for KNN) channel. The best performance data is shown in green while the corresponding changes for the worst channel are outlined in red*

| Channel | | Sensitivity | | | | | | Specificity | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Wake | NREM | | | | REM | Wake | NREM | | | | REM |
| | | | S1 | S2 | S3 | S4 | | | S1 | S2 | S3 | S4 | |
| Fpz-C3 | RF 3 class | 0.9 | | 0.9 | | | 0.7 | 1 | | 0.9 | | | 1 |
| | RF 6 class | 1 | 0.2 | 0.8 | 0.3 | 0.8 | 0.7 | 0.9 | 1 | 0.9 | 1 | 1 | 1 |
| Oz-C4 | KNN 3 class | 0.9 | | 0.9 | | | 0.4 | 0.9 | | 0.8 | | | 1 |
| | KNN 6 class | 0.9 | 0.1 | 0.7 | 0.2 | 0.7 | 0.5 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |
| Cz-C4 | RF 3 class | 0.9 | | 0.9 | | | 0.6 | 0.9 | | 0.9 | | | 1 |
| | RF 6 class | 1 | 0.1 | 0.8 | 0.2 | 0.7 | 0.7 | 0.9 | 1 | 0.9 | 1 | 1 | 1 |
| Cz-Fpz | KNN 3 class | 0.7 | | 0.8 | | | 0.3 | 0.9 | | 0.7 | | | 1 |
| | KNN 6 class | 0.8 | 0.1 | 0.7 | 0.1 | 0.6 | 0.4 | 0.9 | 1 | 0.8 | 1 | 1 | 0.9 |

The ROC plot, shown in figure 5.19, shows an increase in the area under the curve for all classes in comparison to the previous tests. Again, the Wake and S4 classes have the highest area, with the S1 and S3 yielding the lowest. However, the increase in these two classes is also the highest. The REM class is represented well. Unfortunately, this method also fails to solve the issue of adequately representing S1 and S3. The fact that all 3 methods fail to create a solid representation can be explained by looking at the definitions for the stages. The classification of S1 is most commonly mistaken with the REM, S2, and Wake stages. The similarity of S1 with REM is apparent. The waves in both stages are highly similar with the only difference coming from the eye movements in the REM stage. Classification as S2 and Wake can be explained by a combination of two factors: scorer bias and underrepresentation of the S1 stage. The underrepresentation of the stage means that the algorithm might not have enough samples as to accurately formulate the conditions for the S1 stage. The bias comes from the fact that when a human expert scores the epochs, they are doing it sequentially. As it is often the case that Wake and S2 are either preceding or following an S1 stage they might be misclassified by the scorer since they take the previous epoch into account. The same reasoning can explain the shortcomings in the S3 classification. The similarity of the S3 to the S4 stage is apparent with the only difference coming from the percentage of Delta waves. This is a

highly subjective measure when human scorers are involved. Additionally, the S3 stages are also not abundant in the dataset.  It can also be seen that the S2 stage is one of the most common ones to assign to any other stage. This is most likely because of the overrepresentation that the S2 stage has in the dataset.

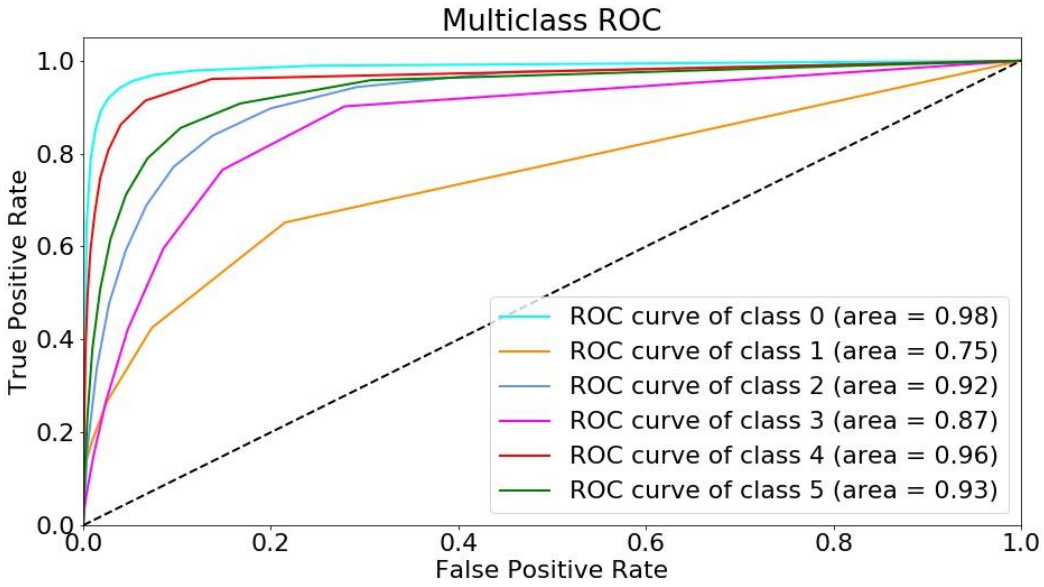



*Figure 5.19. ROC curves for all classes under the Oz-C3 channel for 6 classes using RF. The classes go as follows: 0-Wake ;1-S1;2-S2; 3-S3; 4-S4; 5-REM*

Finally, since there were 30 features involved in the set, the previous method of scoring was unfeasible. Therefore, the method was changed to count only the 7 best features for each channel. From there, the first feature was assigned a score of 7 and the seventh a score of 1. Going over the channels yielded the results shown in figure 5.20. It can be seen that the dominant features come from the Gamma and Beta bands. This can explain the high score of the Wake score since no other sleep stage is characterized by these bands. Additionally, this influence can be attributed to the abundance of the Wake sleep stage in the dataset. Several Theta and Delta band features can be seen as well. Ideally, they would have been the ones with the dominant features, since they are most common in the sleep stages which interest us. It is unexpected that the Alpha waves are not present as dominant at all. The sleep spindles characterizing the S2 stage are in the 10 to 12 Hz range which is the Alpha band. It is unexpected because the S2 is classified correctly most of the time and is well represented in the dataset.

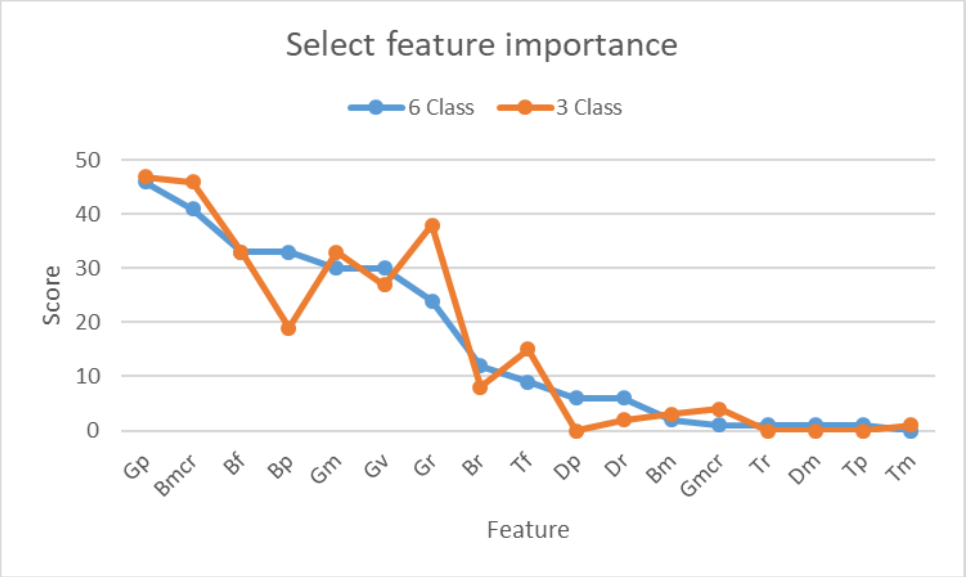



*Figure 5.20. Feature importance for the RF algorithm derived through summing the importance of the first 7 features of each model across all channels.*

## 5.3 Optimization of performance

### 5.3.1 Parallel processing

The time it takes for the computation of converting the .edf files into .csv files with columns of filtered epoch signals is greatly reduced the process is parallelized. The time it takes for the computation of all files for each channel sequentially is 131.1 minutes while using a parallel process takes only 35.6 minutes. This means that it takes almost 3.7 times as much time to do the computation sequentially. The results for each file are displayed in figure 5.21 and show the corresponding times for computation of each file together with the length of the file given in number of epochs. It is clear that parallelizing the process reduces the time significantly.

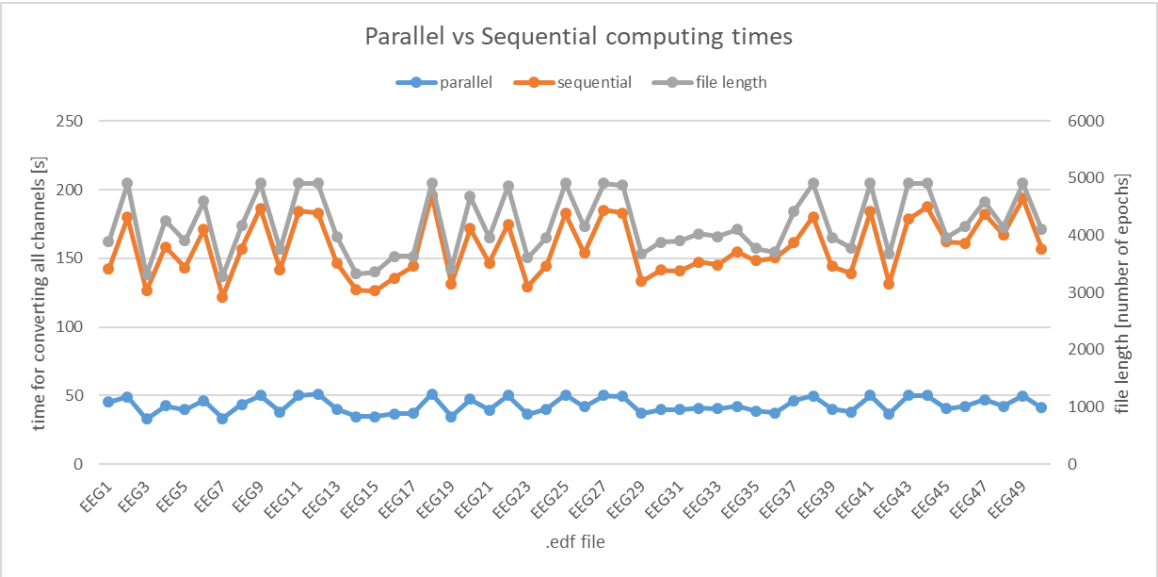



*Figure 5.21. Parallel vs sequential processing of data. Parallel processing is shown in blue at the bottom of the figure while the sequential processing time is shown in orange on top. The second y-axis shows the length of the files being processed given in number of epochs.*

### 5.3.2 ML Algorithm optimization

After all the features have been extracted and evaluated it is wise to see if combining them will yield an even better result than the individual sets. Furthermore, performing dimensionality reduction through PCA can add to the accuracy of the models. Finally, the features are evaluated both by the same method used before (RF feature importance) and by a recursive feature elimination model. The first step was an evaluation of the performance of the channels under all features for the 4 different conditions (RF 3 and 6 classes, KNN 3 and 6 classes). This is displayed in figure 5.22. No significant changes with regards to the CWT spectral features test can be found. Again, the Cz-Fpz and Fpz-C3 channels have the highest overall accuracy (3 class RF) with the Cz-Fpz also having the lowest overall accuracy (6 class KNN). All the changes in comparison to the CWT spectral features are within the cross-validation error. This leads to the conclusion that the CWT spectral features are dominant in these models. Any further analysis is likely to yield very similar results to the ones from section 5.2.3. However, visualizing how the features relate to each other might give an idea of how they complement each other. Therefore, the pair plot of the 10 most dominant features is given in figure 5.23.

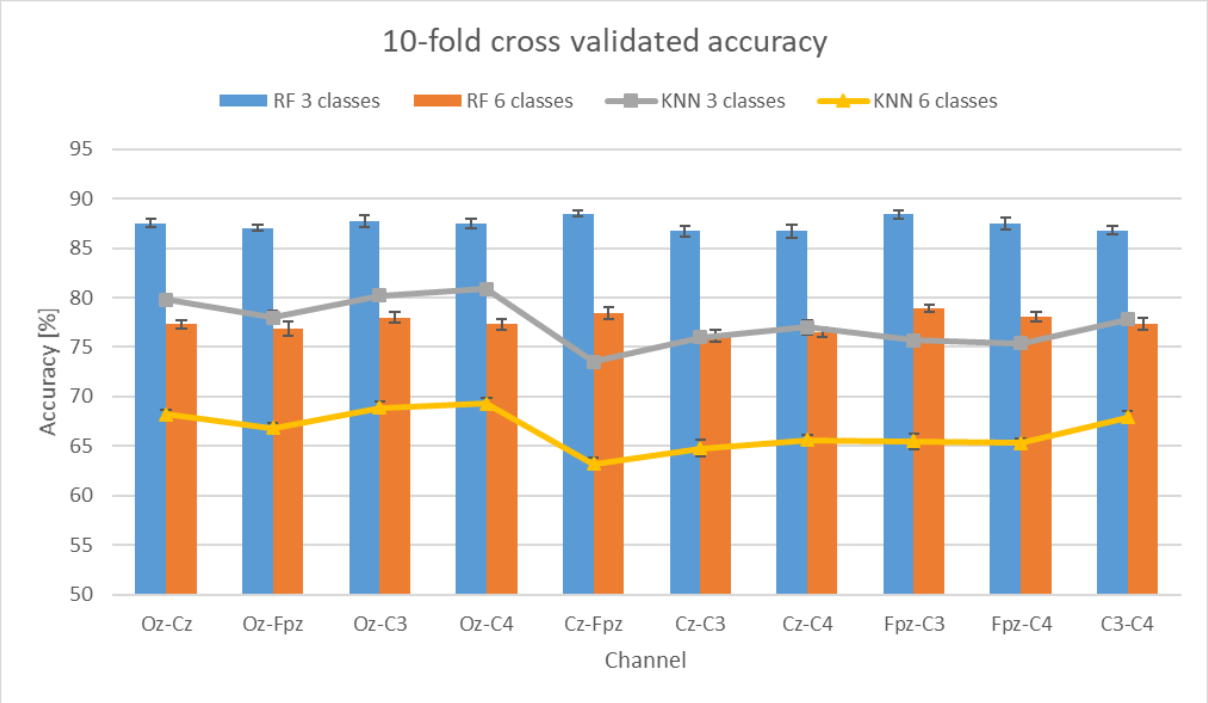



*Figure 5.22. 10-fold cross-validated Accuracy with error ranges for each EEG channel of the 10-second Hut lab data for the combined feature set under both a Random Forest and a K-nearest neighbour (k=30) classifiers for both 3 and 6 classes (top).*

It can be seen that all features are distributed in varying ranges with only the Gamma band features forming distinctive line patterns. Additionally, figure 5.23 shows the ranking of all features using the Random Forest importance calculation. It can be seen that the conclusion about having dominant CWT spectral features is confirmed.

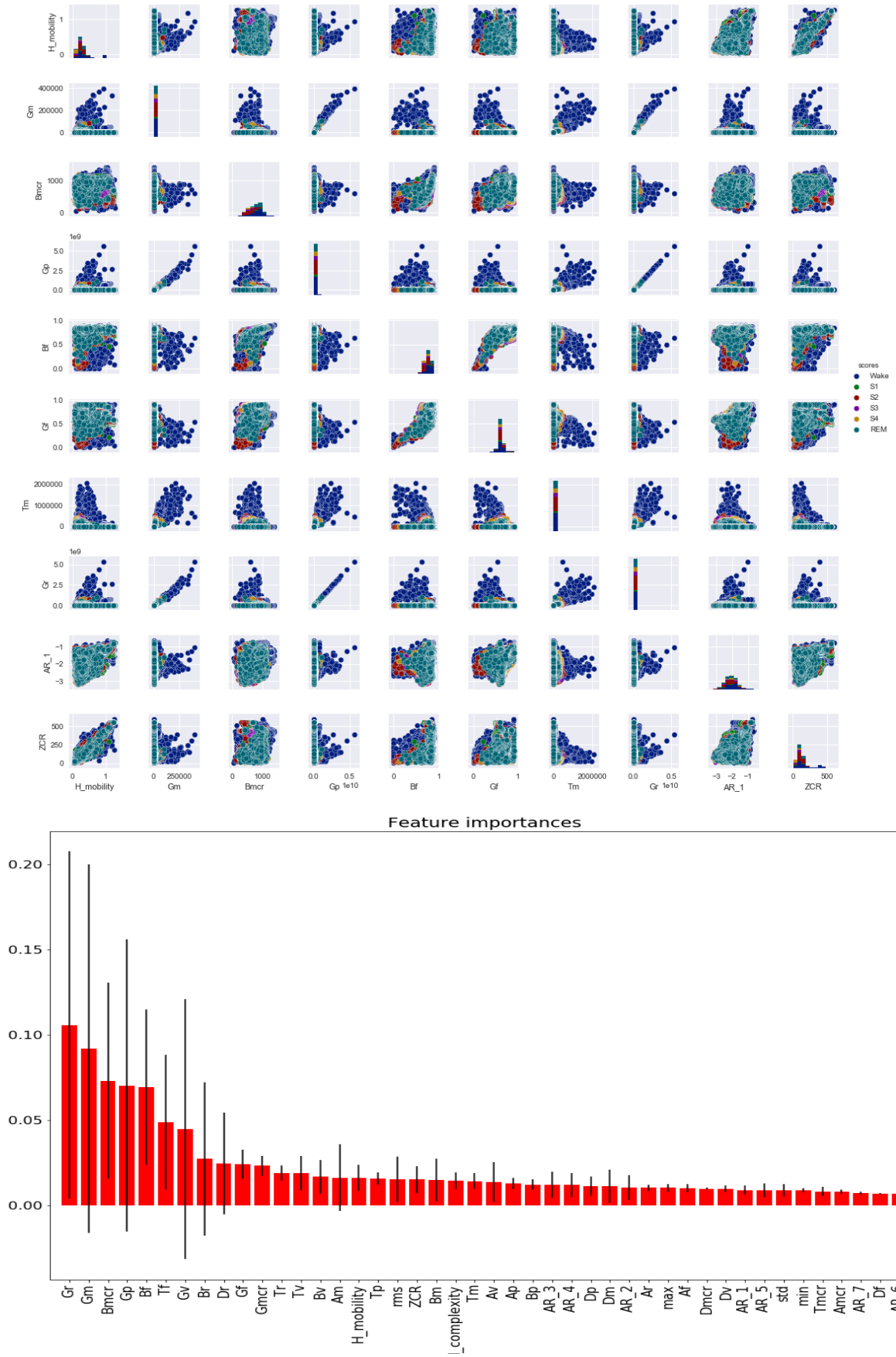



*Figure 5.23. Top: Pair plot of selected important features. Bottom: ranking of features based on their importance to the RF model*

In addition to the method for feature importance evaluation that has been used so far in all tests, an additional recursive feature elimination (RFE) method was used. This method selects the features recursively selecting smaller and smaller sets of features with each iteration by assigning weights to each feature. In the end, the most important ones have the highest weight. With each iteration, the least important features are removed until a selected number of features is reached. In this case, this number is 1 since the interest is in the rank of all features The results are shown in figure 5.24 and it can be seen that even though the positions of the features are not exactly the same, the Gamma and Beta band features are still dominant. The differences in positions are to be expected since each iteration will yield a slightly different result. The general trend, however, remains the same. A curious trend is that the AR features seem to be ranked in the last positions even after some features from the time domain such as Hjorth mobility, complexity, and ZCR. This is unexpected since the AR features gave a better overall accuracy for the models. This can possibly be caused by the fact that the values of the AR features are all relatively small ranging from -1.3 to 0.3. However, the other features can be in the hundreds and thousands. This would mean that while they are suited to be used independently as a set, combining them in a bigger set would decrease their importance. One thing that can be done to avoid that is to scale all features before feeding them in an ML algorithm. Scaling, however, has no effect on the distribution. Therefore, if the feature set has poor distribution it means that scaling will not help. The importance of the scaled features is given in figure 5.25. It can be seen that there is almost no difference in the importance which would mean that the distribution is the main influencing factor in deciding the importance of the features.

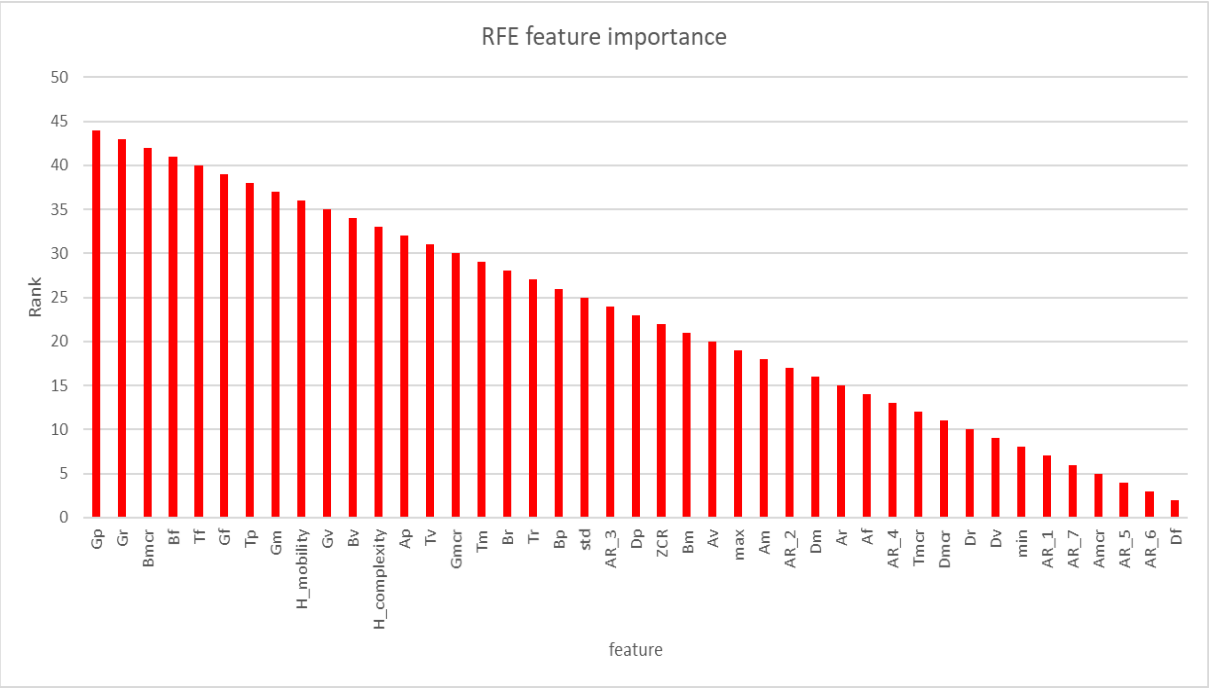



*Figure 5.24. Rank of each feature based on the RFE method.*

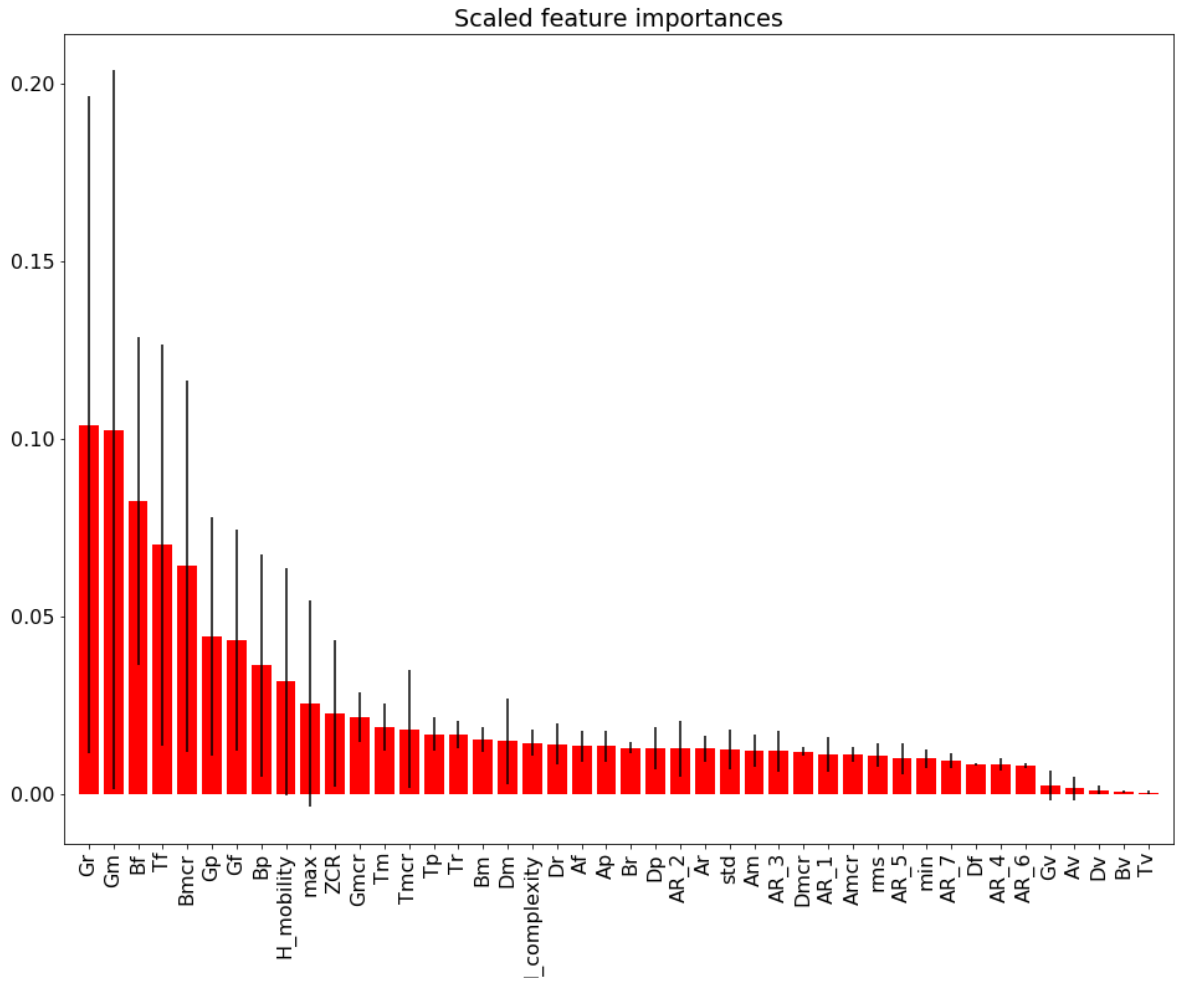



*Figure 5.25. Scaled feature importance of the combined feature set.*

As final optimization procedure, dimensionality reduction can be performed. Again it is important to scale the data before feeding it to a PCA. PCA searches for the dimension which gives the highest variance and therefore having varying scales for the features is undesirable. After scaling, all features are mapped to a uniform scale and therefore the true variance can be seen. The data points from the feature set in regards to the first and second principal components can be seen in figure 5.26. Again it can be seen that the Wake class has the largest variance with some notable data points for the S2 and S4. In the 3 class problem, only the data points for the Wake class can be seen. The ones for the NREM and REM class are likely to have fallen behind the Wake data points and thus not seen on the image.

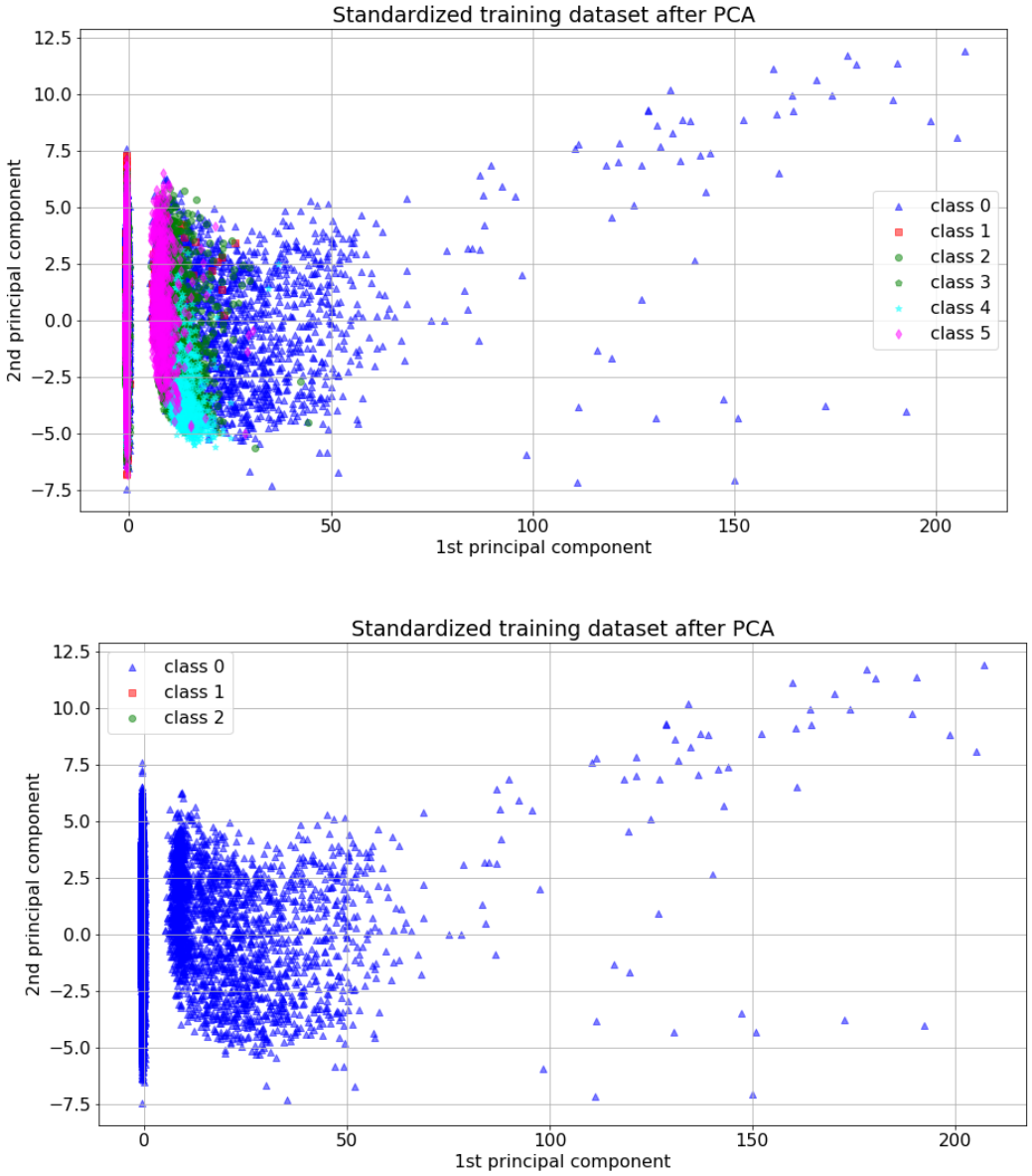



*Figure 5.26. PCA of the Standardized dataset. The classes go as follows: (class 0: Wake, class 1: S1, class 2: S2, class 3: S3, class 4: S4, class 5: REM) in the top image and class 0: Wake, class 1: NREM, class 2: REM. The PCA are given for both the 6 class problem (top) and the 3 class problem (bottom).*

The overall accuracy for the 6 class problem with 44 principal components is 77.3% which is very close to the mean of the model without PCA. The accuracy as a function of the number of principal components can be seen in figure 5.27. It can be seen that the model is at its highest accuracy for the 3 class problem but with no scaling of the input for the PCA at a value of 87.1%. As the number of the principal components is reduced, the accuracy of the model starts to decrease. It can also be seen how the accuracy of the unscaled models is higher when more dimensions are used for the model but after reducing the number of principal components to less than 15, the scaled models start to become more accurate. Unfortunately, PCA does not increase the accuracy. This is probably because the features represent the Wake class with a noticeably higher variance than the rest. It is clear how the data points for all other classes are clustered on the left side of the graph. If it was the case that only this cluster was to be analyzed by the PCA then the accuracy would have increased. However, the outliers of the Wake class have an effect on the performance even after scaling.

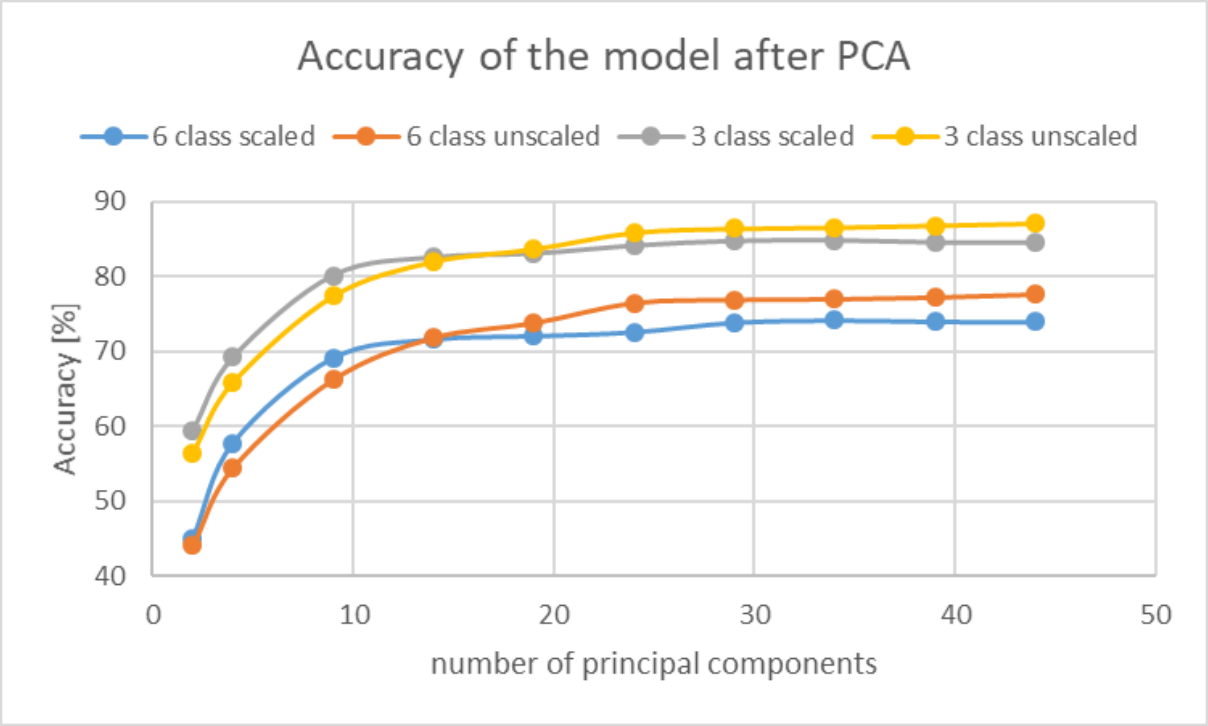



*Figure 5.27. Accuracy of the PCA based model versus the number of principal components.*

## 5.4 ML performance evaluation and comparison

### 5.4.1 HMM comparison and evaluation

A final check of how representative the extracted features are can be done by feeding them in an unsupervised algorithm and evaluating its performance. This is useful because the unsupervised algorithm has no bias. It discovers only patterns in the data. This is a great indicator of how independent the features are from each other. As discussed earlier an HMM is used for this unsupervised classification. The HMM is trained using an EM algorithm (Baum-Welch) to estimate the parameters of the HMM and a Viterbi algorithm to predict the most likely sequence of hidden states. This process was repeated 10 in order to simulate the cross-validation values seen in the previous tests. Using an EM algorithm always runs the risk of settling at a local minimum. This would mean that the model becomes less robust to external data. It also means that it settles to some state which is often not representative of the actual structure underlying the process. Running the EM algorithm 10 times is an attempt to counter this effect. The channel used for this test was the Fpz-C3 since it had the highest accuracy of the combined feature sets. The Accuracy of the HMM is on average 17.74% ± 1.19. For a 6 class problem, this is only very slightly better than guessing (16.6%) if the chance for each class is the same. However, this is not really the case as it can be seen from the confusion matrix as seen in figure 5.28. Even the unsupervised algorithm classifies most of the epochs as Wake. As in the previous tests it can be seen that the second most epochs are attributed to S2 and the third most to REM. The rate of classification of S1, S3, S4 is far lower than chance, meaning that there is a structure in the model. The fact that most epochs are classified either as Wake or S2 means that the hypothesis of having an overrepresentation of these 2 classes is not without basis.

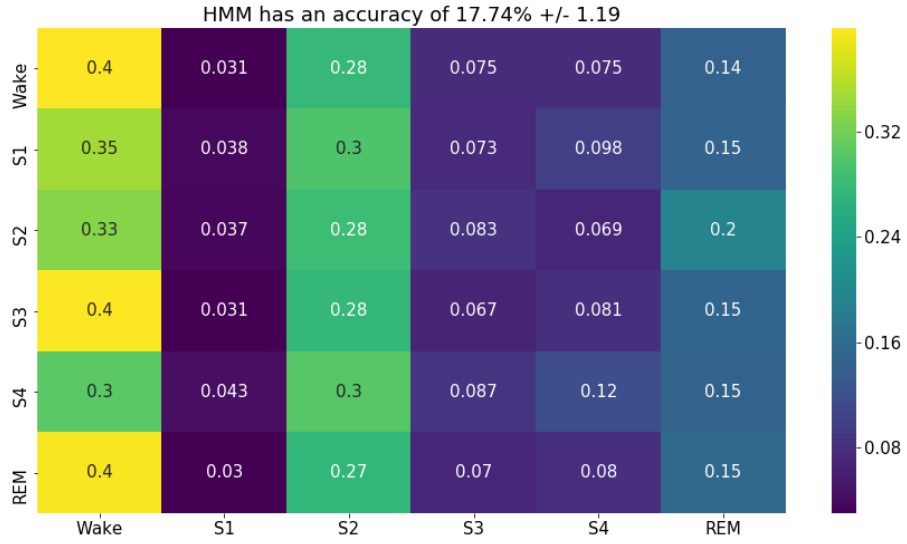



*Figure 5.28. Confusion matrix of the 6 class HMM model.*

The same thing happens when the HMM model is trained to recognize only 3 states. The confusion matrix for the 3 class problem is seen in figure 5.29. There is no change in the classification of REM classes. There is a slight improvement in the NREM classification but still more REM stages are classified as NREM then NREM stages classified as NREM. The overall accuracy is 33.52% ± 1.52. This again is very close to just pure chance if not for the overwhelming Wake and NREM chances.

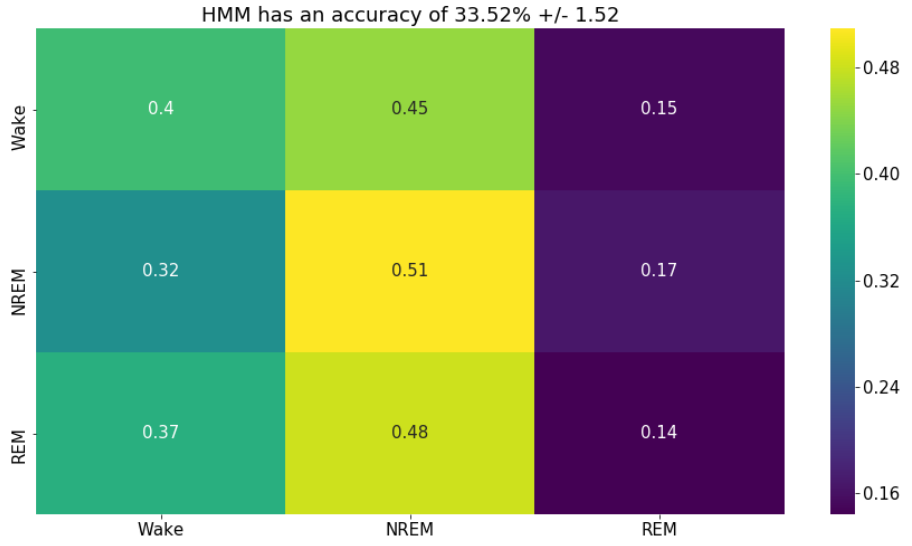



*Figure 5.29. Confusion matrix of the 3 class HMM model.*

Besides looking at the HMM from the purely performance-oriented point of view it is also useful to examine the HMMs themselves. By looking at the state transition matrix. It will be unnecessary to look at the emission matrix since then the values will be given as values for each individual feature. This is not very useful both because there are 44 emissions (44 features) for each class, resulting in 264 individual values. Furthermore, these values are continuous and not discrete therefore it will be difficult to find structure and a link to the features fed into the model. The transition matrix, on the other hand, can be useful when evaluated against the original distribution of sleep stages. The probabilities of switching there should coincide with the number of epochs from each stage.

Therefore, the probabilities for switching to S1 should be smallest and the probabilities of switching to Wake, the highest.

The first thing of note in the transition matrix for the 6 class problem, shown in table 5.4, is the fact that there is an overwhelming tendency of the model to remain in the state it is currently in. That can be confirmed by looking at the actual transformation matrix that can be extracted from the raw hypnogram data seen in table 5.5. The table was extracted by counting how many times each state changes to one of the others. It is clear that the same tendency of remaining in the previous state is also present. Even though the numbers are not the same it can be seen that the HMM manages to capture an important aspect of the structure in the data. Another thing of note is that the transition probability for some state transitions is 0 (Wake-S4, S1-S4, REM-S3, REM-S4). This might suggest that the model is not to be assumed to be ergodic (fully connected). However, it is only the case of this data set that these connections between stages are missing. If a robust model, capable of handling diverse independent data is to be created, this assumption should be approached with care.

*Table 5.4. Transition matrix given by the 6 class HMM.*

| Sleep stage | Wake | S1 | S2 | S3 | S4 | REM |
|---|---|---|---|---|---|---|
| **Wake** | **0.83** | 0.02 | 0.03 | 0.04 | 0.04 | 0.03 |
| **S1** | 0.04 | **0.74** | 0.04 | 0.06 | 0.05 | 0.06 |
| **S2** | 0.06 | 0.05 | **0.71** | 0.04 | 0.07 | 0.06 |
| **S3** | 0.06 | 0.06 | 0.04 | **0.74** | 0.05 | 0.05 |
| **S4** | 0.06 | 0.06 | 0.07 | 0.05 | **0.71** | 0.04 |
| **REM** | 0.05 | 0.06 | 0.07 | 0.05 | 0.04 | **0.73** |

*Table 5.5. Transition matrix given by the distribution of the original data for the 6 class problem.*

| Stages | Wake | S1 | S2 | S3 | S4 | REM |
|---|---|---|---|---|---|---|
| **Wake** | **0.995609** | 0.00305 | 0.000986 | 6.57367E-05 | 0 | 0.000289 |
| **S1** | 0.008103 | **0.896899** | 0.089969 | 0.001676446 | 0 | 0.003353 |
| **S2** | 0.002314 | 0.004662 | **0.958659** | 0.029891628 | 0.000527 | 0.003947 |
| **S3** | 0.002746 | 0.002683 | 0.098103 | **0.854250495** | 0.041068 | 0.00115 |
| **S4** | 0.002084 | 0.001746 | 0.005407 | 0.02861327 | **0.962037** | 0.000113 |
| **REM** | 0.001868 | 0.004949 | 0.002459 | 0 | 0 | **0.990724** |

The same observation can be made for the 3 class problem. In this case, the tendency to remain in the same state is stronger (>90%). Moreover, here no missing connections can be seen, which make sense if compared to table 5.5.

*Table 5.6. Transition matrix given by the 3 class HMM model.*

| Sleep stage | Wake | NREM | REM |
|---|---|---|---|
| **Wake** | **0.9** | 0.05 | 0.05 |
| **NREM** | 0.07 | **0.85** | 0.08 |
| **REM** | 0.08 | 0.06 | **0.86** |

*Table 5.7. Transition matrix for the 3 class problem given by the original distribution of the data.*

| Sleep stage | Wake | NREM | REM |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **Wake** | **0.995609** | 0.004102 | 0.000289 |
| **NREM** | 0.002758 | **0.994464** | 0.002778 |
| **REM** | 0.001868 | 0.007408 | **0.990724** |

### 5.4.2 Comparison with other studies

In order to have an unbiased evaluation of the performance of the algorithm, a comparison with studies that have the same topic is needed. These studies vary in the methods used, the number of sleep stages considered, the resources used and also in general purpose for development. Table 5.8 shows the sleep stages falling under the variable for the corresponding number of classes.

**Another way of evaluating the comparison of the**

*Table 5.8. Number of classes and the corresponding sleep stages they include.*

| # Classes | Sleep Stages |
|---|---|
| **6** | Wake, S1, S2, S3, S4, REM |
| **5** | Wake, S1, S2, SWS(S3-S4), REM |
| **4** | Wake, S1-S2, SWS(S3-S4), REM |
| **3** | Wake, NREM, REM |
| **2** | Wake, Sleep |

Table 5.9 shows the performance of some of the state of the art algorithms given in overall accuracy. It is immediately apparent that the accuracy of the proposed algorithm is not the highest. This is due to the varying conditions of each study. In all studies, fewer subjects were used resulting in a lower total number of epochs. The results for the 6 class problem and the 3 class problem are bolded. While the 6 class models outperform the proposed model, the 3 class models have similar results across all studies.

*Table 5.9. Accuracy comparison with other studies. In blue are the scores for 6 classes and in orange for 3 classes.*

| Authors | # subjects | Method | # classes | Accuracy [%] | Year |
|---|---|---|---|---|---|
| **Huang et al. [42]** | 10 | SVM | 4 | 77.12 | 2013 |
| **Berthomier et al. [43]** | 15 | Fuzzy Classification | 5, 4, 3, 2 | 71.2, 74.5, 88.3, 95.4 | 2007 |
| **Hsu et al. [44]** | 8 | FNN, PNN | 5 | 87.2 | 2013 |
| **Liang et al. [45]** | 20 | LDA | 5 | 83.6 | 2012 |
| **Hassan et al. [46]** | 8 | Ensemble bagging | 6, 5, 4, 3, 2 | 85.57, 86.53, 87.49, 89.77, 95.05 | 2015 |
| **Zhu et al. [47]** | 8 | SVM | 6, 5, 4, 3, 2 | 87.5, 88.9, 89.3, 92.6, 97.9 | 2014 |
| **Proposed** | 50 | RF, KNN | 6, 3 | 79.3, 88.8 | 2018 |

### 5.4.3 Performance on a separate database

Another way of evaluating the performance of the model and the quality of the features is through checking how well it classifies data from a separate database. This evaluation is aimed at determining how well the algorithm works under unseen and independent data from a different source and extracted under different conditions. In a sense, this is an evaluation of robustness.

The DREAMS subject database has a total number of 121618 epochs with a distribution shown in figure 5.30. In this database, the S2 stages are much more abundant with the S1 and S3 being the least represented as with the Hut dataset. Before training a model the unknown sleep stages are again removed in order to have consistency with the Hut dataset.

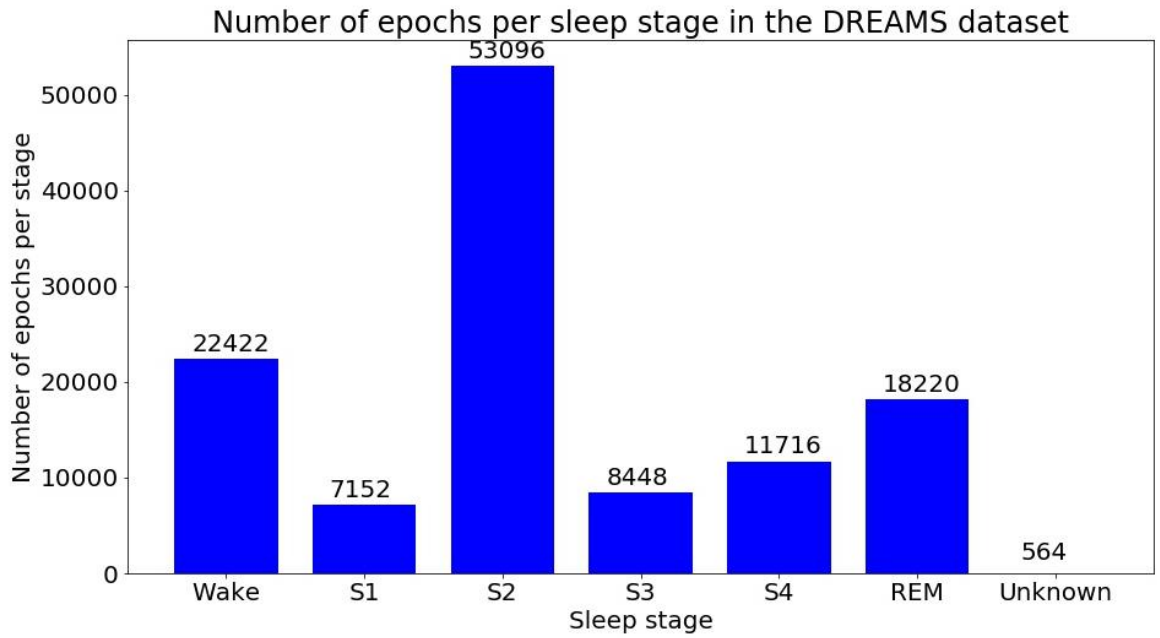



*Figure 5.30. DREAMS subject database sleep stage distribution given in number of epochs.*

After the parameters of the DREAMS dataset have been made consistent with the Hut dataset, 4 models were tested. As in the previous tests, both a 3 class and a 6 class RF and KNN models were trained. The accuracy scores can be seen in figure 5.31. The channel used from the DREAMS dataset is Cz-Fp1 which is the closest to the one with both the highest (for the RF) and the lowest (KNN) performance in the single-set tests. This was chosen because it is indicative of the worst case scenario, in regards to robustness, which gives more insight into the real robustness of the model. This is why the highest 10 fold cross-validated accuracy, in this case, falls on the Cz-Fpz channel. It is apparent that the accuracy across all channels has dropped slightly. This is to be expected since the data have slightly different characteristics and that might resonate in the values for the features. However, the models still have a high overall performance, reaching 83.9% accuracy at the highest, considering that the study on overall scorer agreement done by the AASM described in Chapter 1 reported an overall accuracy of 82.6% between scorers. It can also be seen how the results from the cross-validation have a much higher variation than in the previous tests. Another thing of note is how the highest 6 class RF accuracy is also higher than the lowest 3 class KNN accuracy (Cz-Fpz).

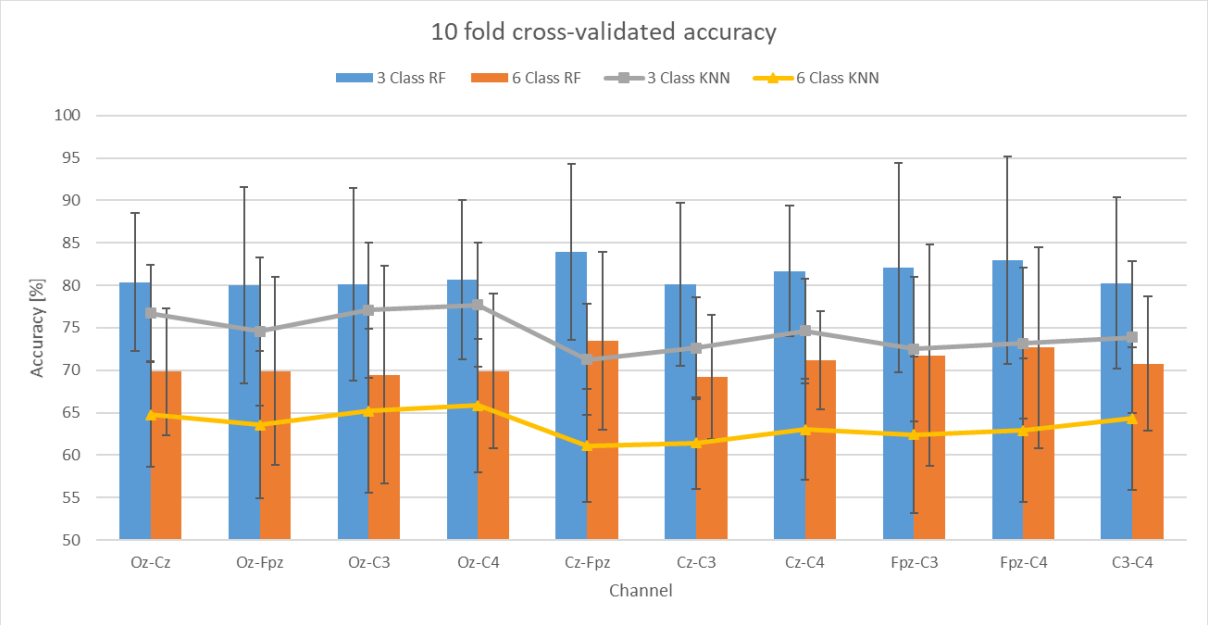



*Figure 5.31. 10 fold cross-validated accuracy for all models based on the Hut lab dataset training data and DREAMS dataset testing data.*

By looking at the confusion matrices (figure 5.32) it becomes apparent that the overall accuracy can be deceiving. It is obvious that there are multiple misclassifications in both the 6 class and the 3 class models. The only sleep stage that has been correctly classified is the S2 (NREM in the 3 class model). Additionally, the S3 class is more often classified as S4 than as itself. The Wake class is classified as REM, S1, and S3 more often than as itself. The S1 is more often classified as REM than as itself. It is also apparent that the S2 dominance from the 6 class model translates into an NREM dominance in the 3 class problem with more than 90% of the epochs being classified as NREM in both Wake and REM.

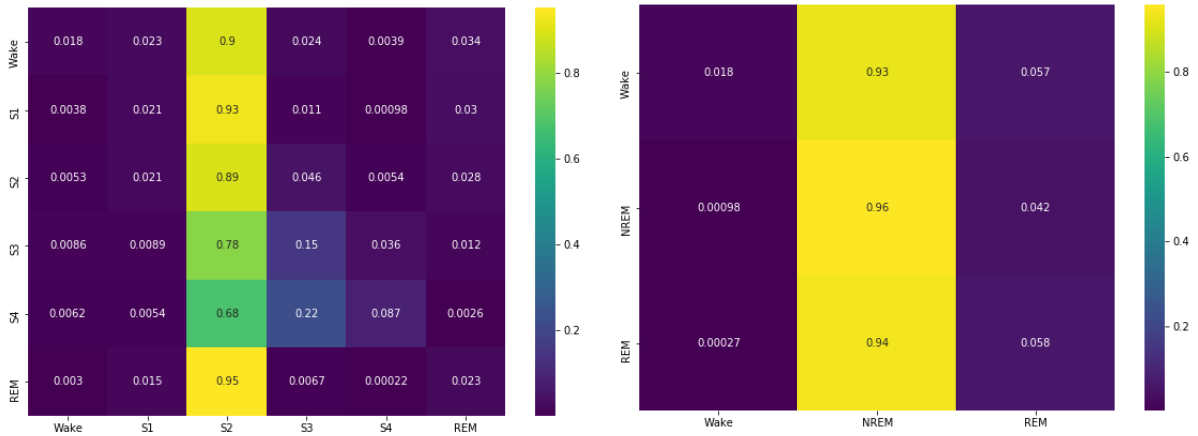



*Figure 5.32 Normalized confusion matrix for the 6 class (left) and the 3 class (right) model for the cross-dataset test.*

By looking at the Sensitivity and Specificity of the model given in table 5.10, it can be confirmed that the S1 and Wake are the least recognized classes. With both having a Sensitivity of 0 in the best cases and a Specificity of 1. The values are also on the low end for S3, S4, and REM with only the S2 class having a high value. In contrast to the previous tests, here the lowest accuracy channels do not always have a lowered Sensitivity and Specificity value. A noticeable improvement in is the

Specificity of the S2, S3 and S4 stages for the Cz-C3 channel. However, the improvement in these classes comes at a cost of a decrease in the values in the Wake and REM stages.

*Table 5.10. Sensitivity and Specificity of the cross-dataset model. In green are the values for the channels with the highest accuracy (Cz-Fpz for RF and OZ-C4 for KNN). In red are the decreases in the values for the channels with the lowest accuracy. In yellow are the increases in values for the channels with the lowest accuracy.*

| Channel | | Sensitivity | | | | | | Specificity | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Wake | NREM | | | | REM | Wake | NREM | | | | REM |
| | | | S1 | S2 | S3 | S4 | | | S1 | S2 | S3 | S4 | |
| Cz-Fpz | RF 3 class | 0 | 1 | | | | 0.1 | 1 | 0.1 | | | | 1 |
| | RF 7 class | 0 | 0 | 0.9 | 0.2 | 0.1 | 0 | 1 | 1 | 0.1 | 0.9 | 1 | 1 |
| Oz-C4 | KNN 3 class | 0.5 | 0.1 | | | | 0.6 | 0.8 | 1 | | | | 0.3 |
| | KNN 7 class | 0.5 | 0 | 0 | 0 | 0 | 0.6 | 0.8 | 1 | 1 | 1 | 1 | 0.2 |
| Cz-Fpz | KNN 3 class | 0.3 | 0.3 | | | | 0.7 | 0.9 | 0.9 | | | | 0.4 |
| | KNN 7 class | 0.3 | 0 | 0.1 | 0 | 0.1 | 0.8 | 0.9 | 1 | 0.9 | 1 | 1 | 0.3 |
| Cz-C3 | RF 3 class | 0 | 0.9 | | | | 0 | 1 | 0.1 | | | | 1 |
| | RF 7 class | 0 | 0 | 0.6 | 0.4 | 0.5 | 0 | 1 | 1 | 0.3 | 0.8 | 0.9 | 1 |

Even though the confusion matrix and the Sensitivity and Specificity table show how the performance of the model is low on every class besides S2, the overall accuracy is high at 83.9%. This is most likely due to the overrepresentation of the S2 stage, leading to a skewness in the dataset and consequently performance. However, it is also worth noting that a model trained on the DREAMS dataset only does not perform poorly. The accuracy for the Cz-Fp1 channel for the 3 class problem is 81.8% +/- 0.9 and for the 6 class problem 68.9 +/- 0.7 with confusion matrices shown in figure 5.33. It can be seen how the classification for Wake, S4, and REM has improved in comparison to the cross-dataset model. However, the misclassifications in S1 and S3 remain more abundant than the correct classifications, with S1 being mistaken with the REM, S2 and Wake stage and the S3 with S2 and S4. This suggests that the performance of the cross-dataset model is not only due to the fact that the signals vary between the datasets but also that the signals in the DREAMS dataset are harder to cluster.

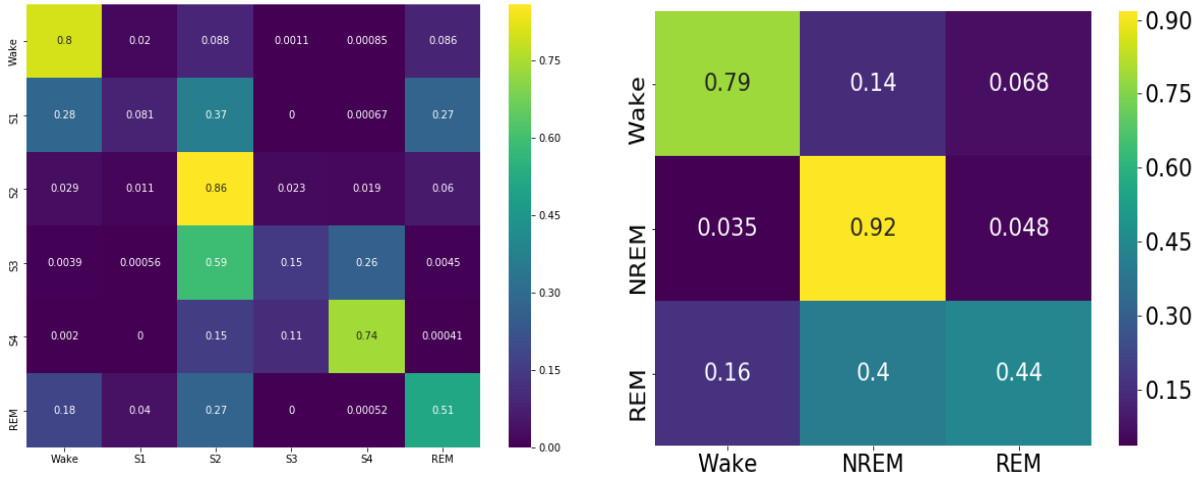



*Figure 5.33. Normalized confusion matrices for both the 6 class (left) and the 3 class (right) RF models of the Cz-Fp1 channel of the DREAMS dataset.*

# 6. CONCLUSIONS & RECMMENDATIONS

This thesis has addressed the research question from the Rationale by providing a detailed analysis of the problem of classifying sleep stages based only on a single EEG channel by means of ML algorithms.

The answer to the first sub-question: “Which features are yielding the highest segregation between the classes?” can be formulated as follows. From the attempted signal processing techniques aimed at feature extraction, the spectral features extracted through Continuous Wavelet Transform and statistical analysis of the separate bands yielded the highest accuracy for all ML models. The progression of features, from time to spectral parametric to CWT spectral, showed continuous improvement, most noticeably in the REM sleep stage. Additionally, the singular features with the highest importance are the Beta and Gamma band features, the Hjorth parameters, the Zero-crossing rate and the third coefficient of the parametric features.

It is unfortunate that all feature sets fail to characterize the S1 and S3 classes sufficiently well, such that they have a higher correct classification rate than a misclassification rate as seen from the confusion matrices. Therefore, it is recommended that that more descriptive features are explored. Such features can be based on the entropy of the EEG signals, their dimensionality or even further spectral band features.

The answer to the second sub-question: “Which EEG channel yields the best performance?” is the Cz-Fpz channel in the highest accuracy model. However, this does not fully cover the spectrum of all the performed tests. In the case of a KNN classifier, the Cz-Fpz channel has the lowest accuracy. For the time domain and the spectral parametric feature sets with RF classifier, the channels with the highest accuracy involve the Oz, C3 and C4 electrodes. However, for the CWT spectral set and the combined set, the highest accuracy involves the Fpz electrode. In the case of the KNN classifier, the highest accuracy comes from electrodes Oz, C3, and C4 for all feature sets.

Unfortunately, this conclusion does not bode well for the development of a comfortable wearable device since the positioning of the Oz electrode especially is where the subject would naturally rest their head. This could be both uncomfortable and interfere with the unobstructed signal of the device.

It can also be concluded that the Random Forest classifier has a better performance than the other two attempted methods, namely KNN and HMM. The RF classifier models performed better than the KNN in the time domain feature set model, the CWT spectral features set model and the combined features set model. The only test in which KNN outperformed RF was the parametric spectral feature set model. However, it must be noted that the performance of the KNN heavily depends on the number of k-neighbors selected.

One recommendation would be to train more models based on varying classifiers such as Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), Gaussian Mixture Models (GMM), Naïve Bayes (NB), and Artificial Neural Networks (ANN).

An additional conclusion is that the sensitivity and specificity values largely follow the same trend in the classifier as the accuracy. In all cases, however, the specificity of the models is too high and the sensitivity too low. Unfortunately, looking at the one-versus-all ROC plots does not suggest a solution to this problem. In a multiclass classifier (non-binary) it can be difficult to adjust the threshold of the classifier such that it satisfies the needs of all classes.

Another thing of note is that the KNN models have a lower training time than the RF, which is a characteristic useful for real-time applications. In regard to further optimization, it can be concluded

that combining the feature sets does not yield higher accuracy than the CWT spectral feature set. Additionally, the use of dimensionality reduction techniques such as PCA does not yield a higher accuracy either. However, using parallel processing greatly reduces the computing time for feature calculation and .edf file conversion.

In addition, the HMM model trained based on the combined feature set did not yield higher accuracy than the supervised models. It will be premature to rule out the HMM as an adequate model for the sleeping brain, however, since the HMM model was trained based on the combined feature set, which fails to characterize the S1 and S3 stage sufficiently well. Additionally, the HMM model did pick up the trend of the process to remain in a state once it has entered it. Therefore, it is recommended that a new HMM is trained if a new feature set is created which describes each class appropriately.

The cross-dataset model shows that training a model on one data set and testing on another does not yield high accuracy. Even though one of the functions of the features is to standardize the representation of the data, they can be skewed by the varying characteristics of that data. For example, varying epoch length (10 and 5 seconds) and varying sampling rate (128 Hz and 200 Hz) can have an impact on the values of the features.

Finally, both dataset used (Hut lab and DREAMS subject) have an overrepresentation of some classes. This affects the sensitivity of the models and their overall accuracy. Additionally, the number of subjects and epochs in the main dataset (Hut lab) is far greater than the ones used in other studies. Even so, the methods proposed in this study are on par with the 3 class models of state of the art models proposed elsewhere.

**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] S. M. Kay, “Modern Spectral Estimation: Theory and Application,” in *Signal*

*Processing. Prentice-Hall*, 1988.

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

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

[38] 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.

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

[40] K. Asanovic, B. C. Catanzaro, D. Patterson, and K. Yelick, "The Landscape of Parallel Computing Research : A View from Berkeley," 2006.

[41] P. Mishra, "A laymans introduction to principal component analysis," 2017. [Online]. Available: https://hackernoon.com/a-laymans-introduction-to-principal-components-2fca55c19fa0.

[42] C. S. Huang, C. L. Lin, L. W. Ko, S. Y. Liu, T. P. Sua, and C. T. Lin, "A hierarchical classification system for sleep stage scoring via forehead EEG signals," *Proc. 2013 IEEE Symp. Comput. Intell. Cogn. Algorithms, Mind, Brain, CCMB 2013 - 2013 IEEE Symp. Ser. Comput. Intell. SSCI 2013*, pp. 1–5, 2013.

[43] C. Berthomier *et al.*, "Automatic analysis of single-channel sleep EEG: Validation in healthy individuals," *Sleep*, vol. 30, no. 11, pp. 1587–1595, 2007.

[44] Y. L. Hsu, Y. T. Yang, J. S. Wang, and C. Y. Hsu, "Automatic sleep stage recurrent neural classifier using energy features of EEG signals," *Neurocomputing*, 2013.

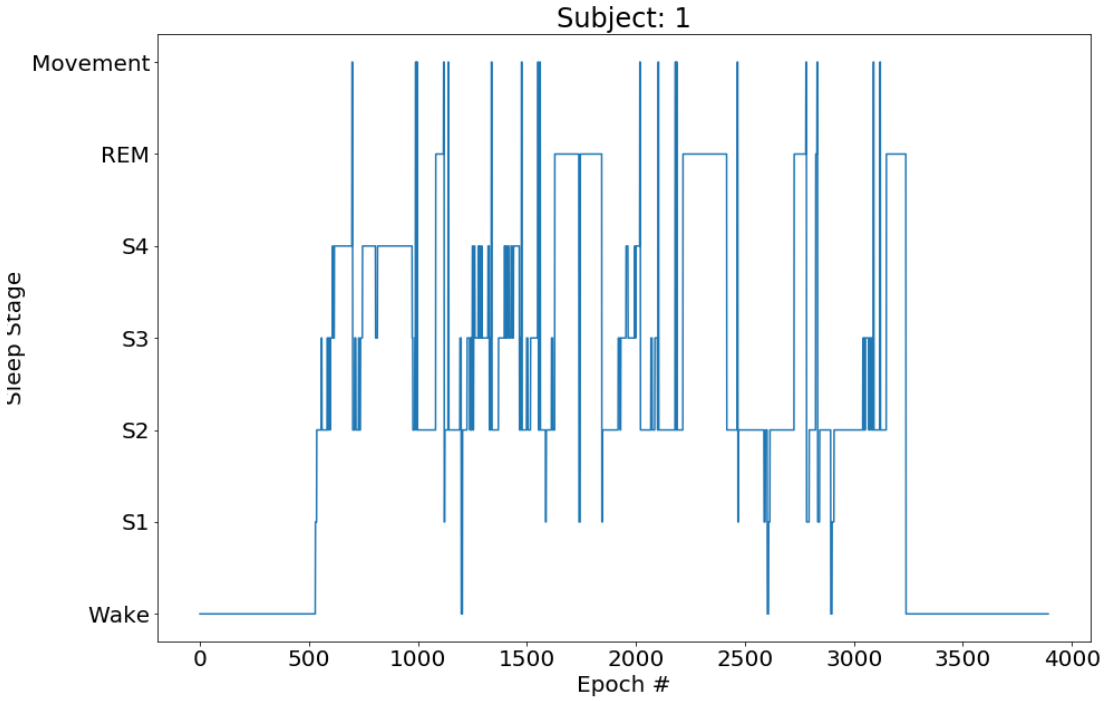
[45] S. F. Liang, C. E. Kuo, Y. H. Hu, Y. H. Pan, and Y. H. Wang, "Automatic stage scoring of single-channel sleep EEG by using multiscale entropy and autoregressive models," *IEEE Trans. Instrum. Meas.*, 2012.

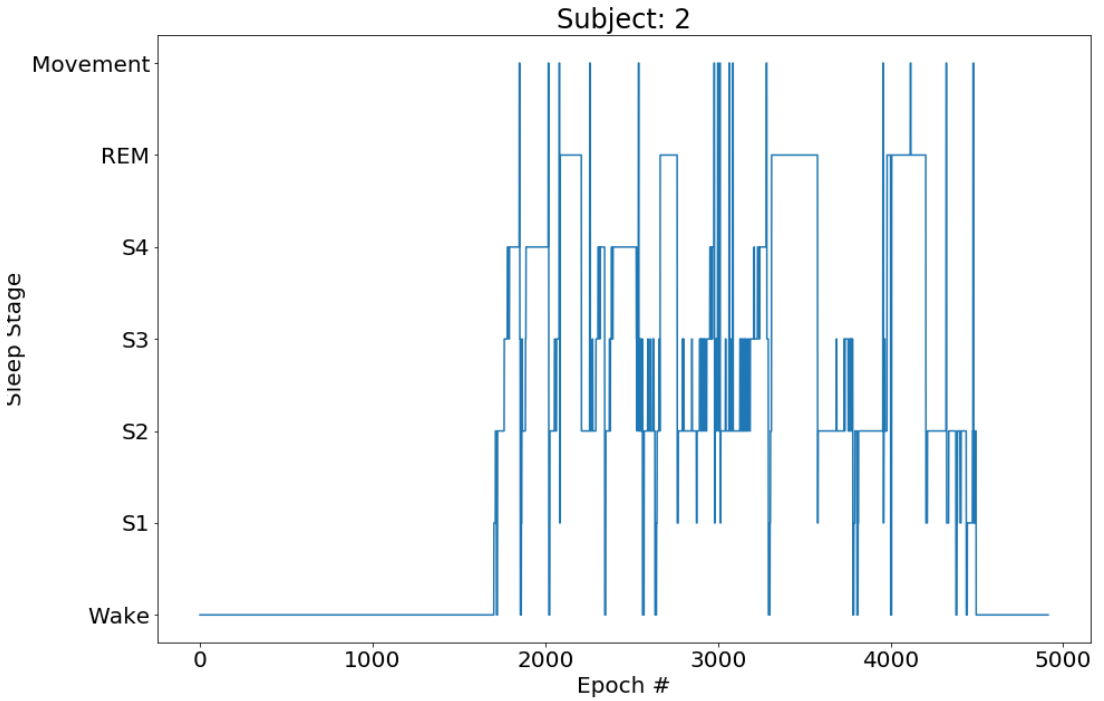
[46] A. R. Hassan, S. K. Bashar, and M. I. H. Bhuiyan, "On the classification of sleep states by means of statistical and spectral features from single channel Electroencephalogram," *2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015*. pp. 2238–2243, 2015.

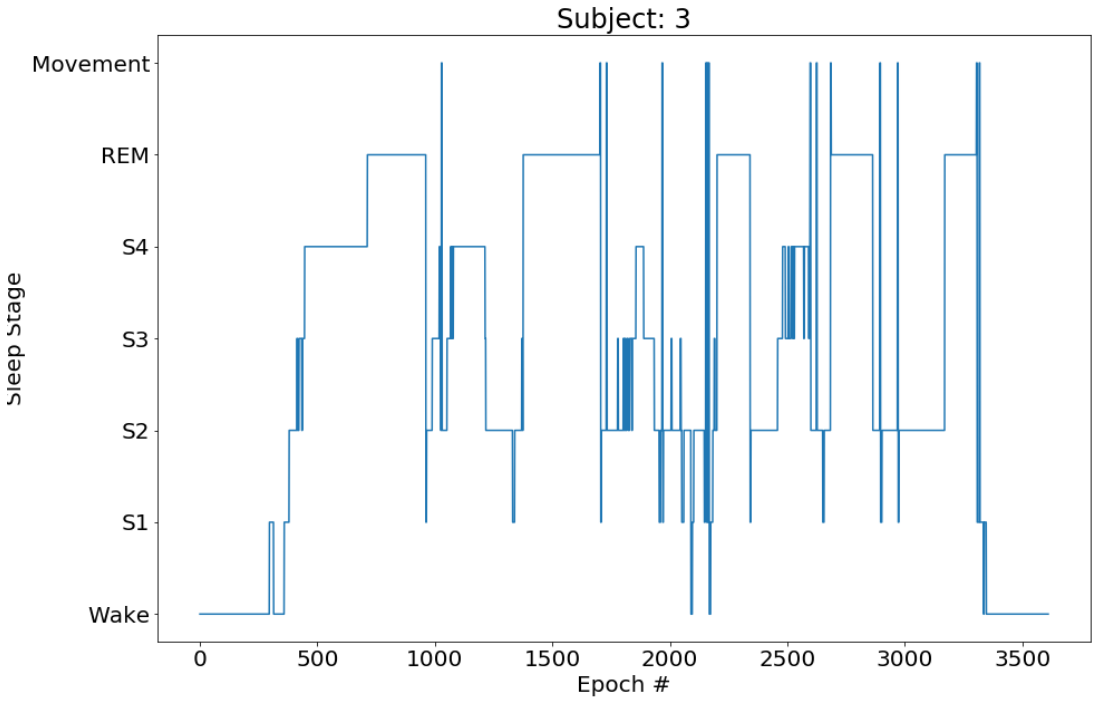
[47] G. Zhu, Y. Li, and P. P. Wen, "Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal," *IEEE J. Biomed. Heal. Informatics*, 2014.

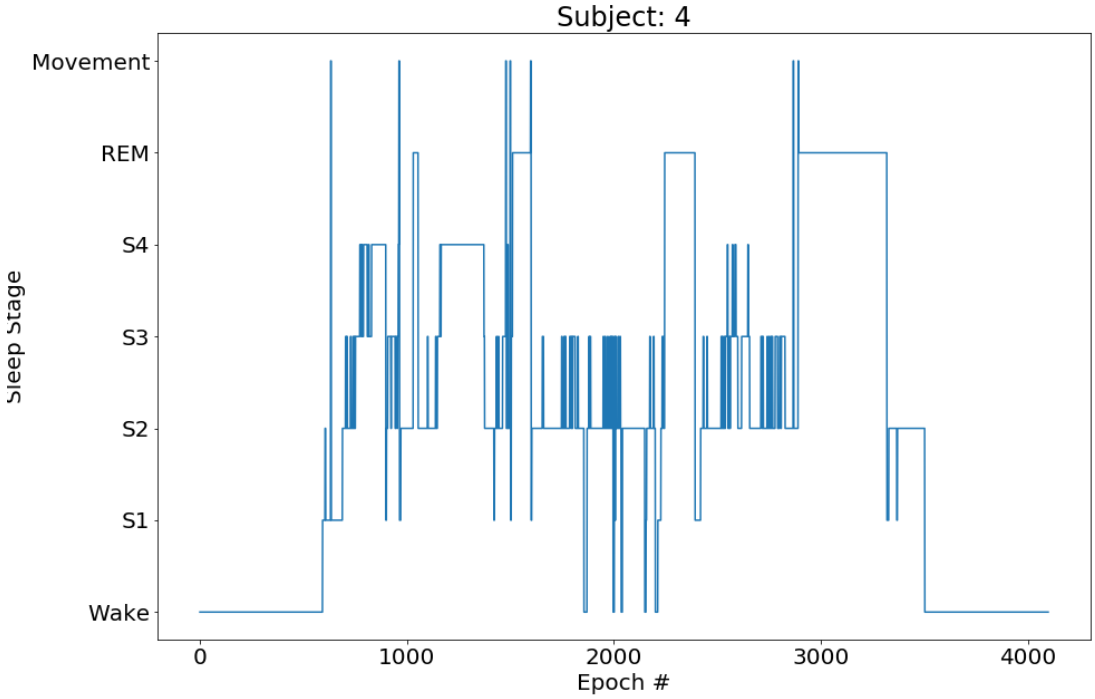
## APPENDIX A

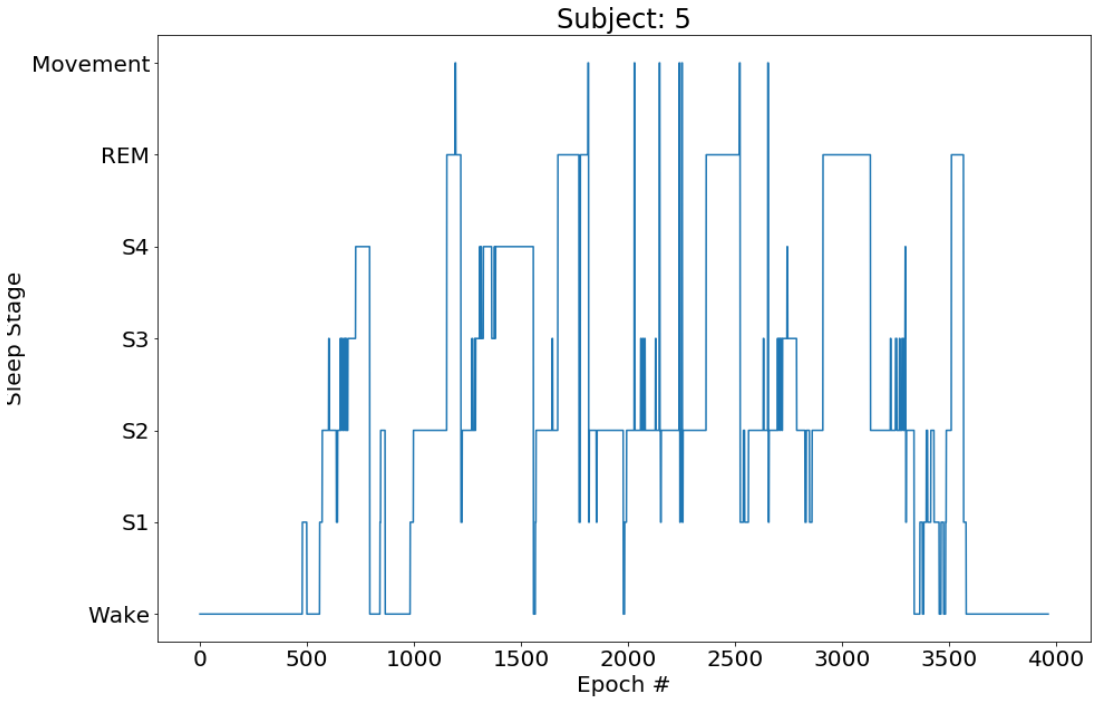
Hypnograms for all subject. The subject number is given in the title.

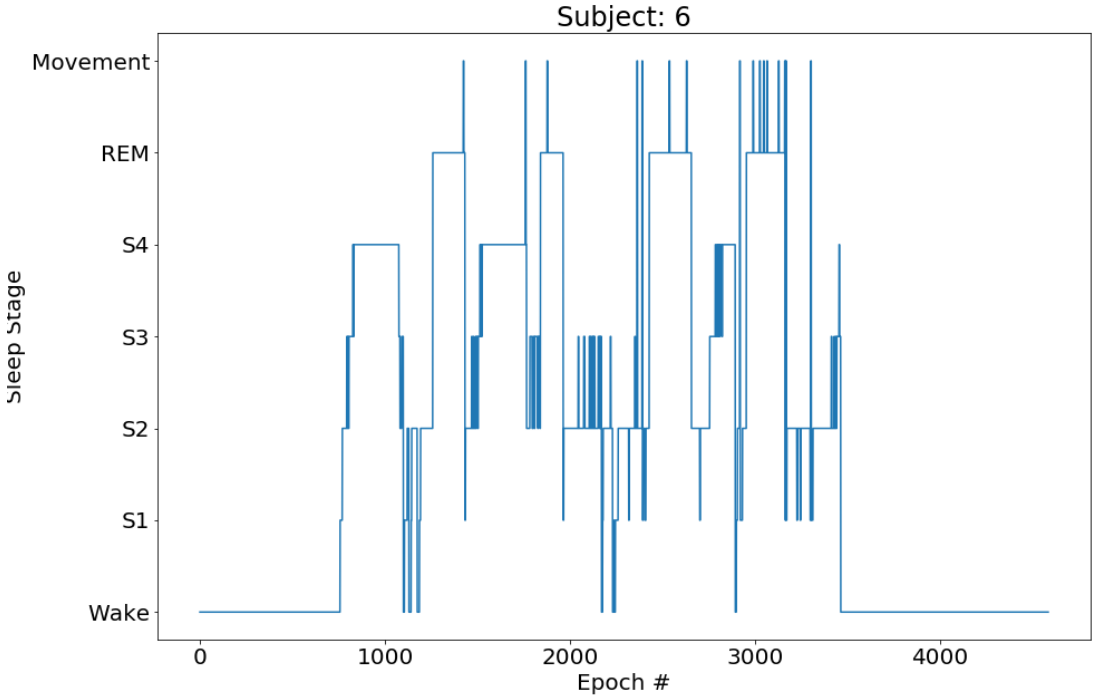


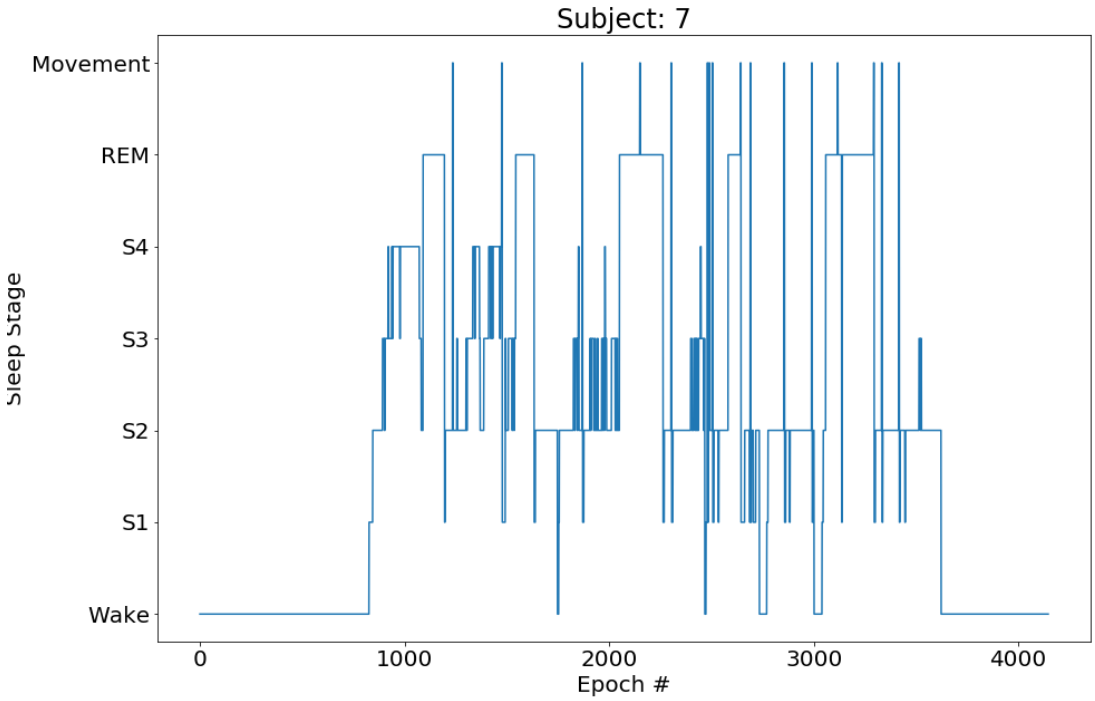



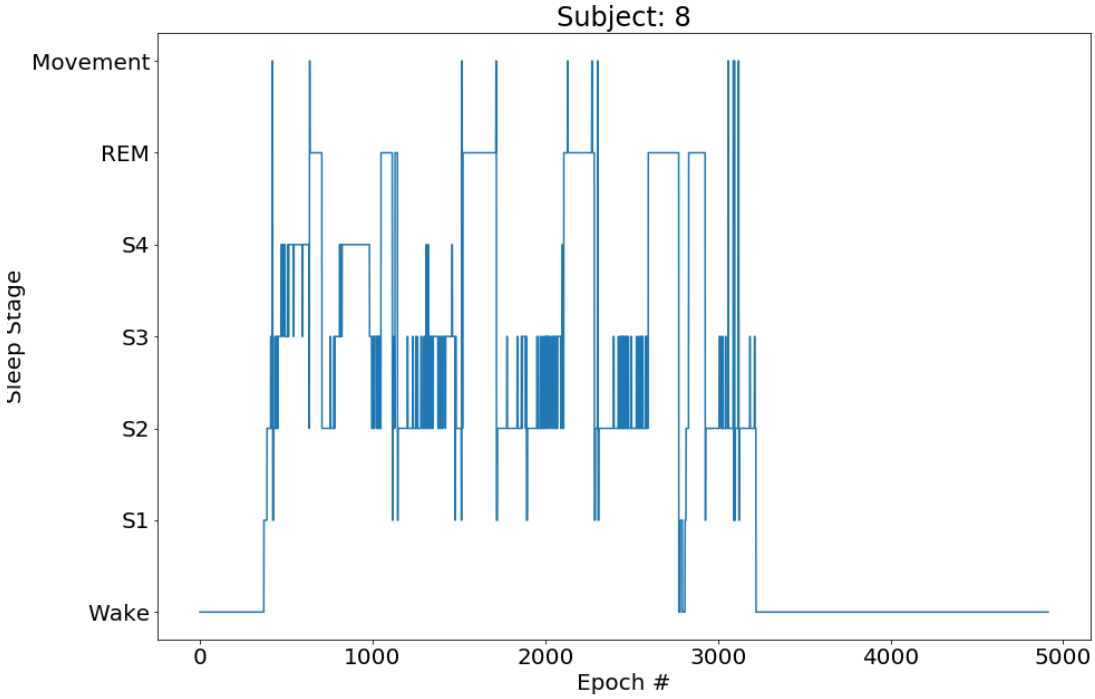

Subject: 3
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #

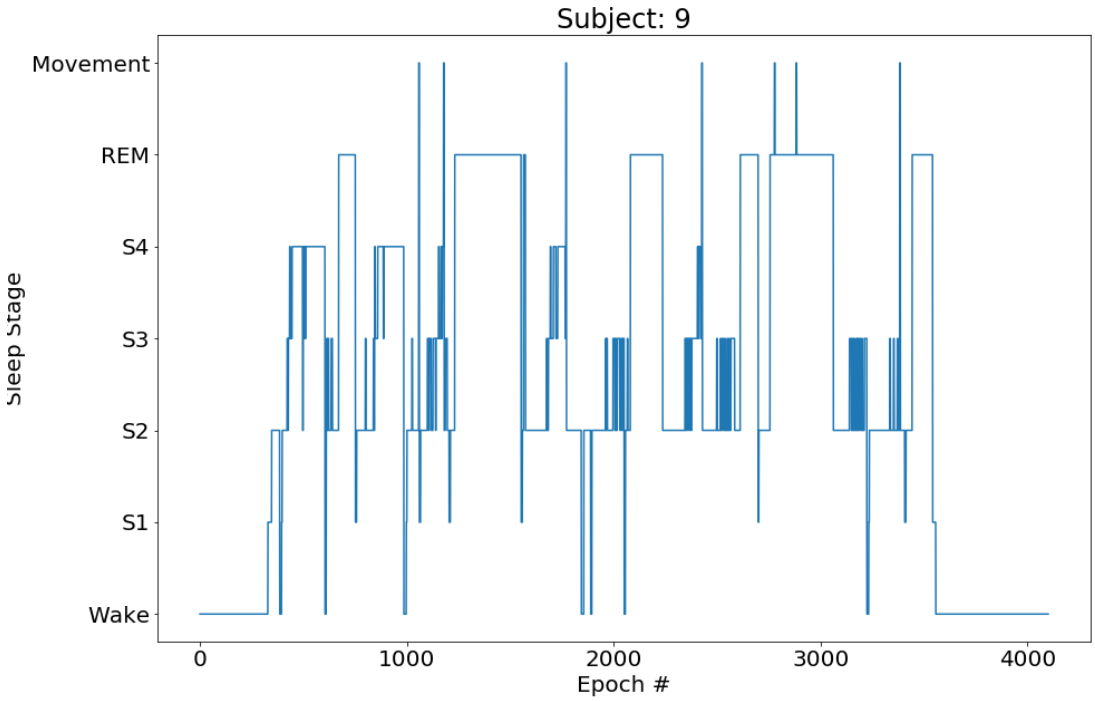

Subject: 4
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

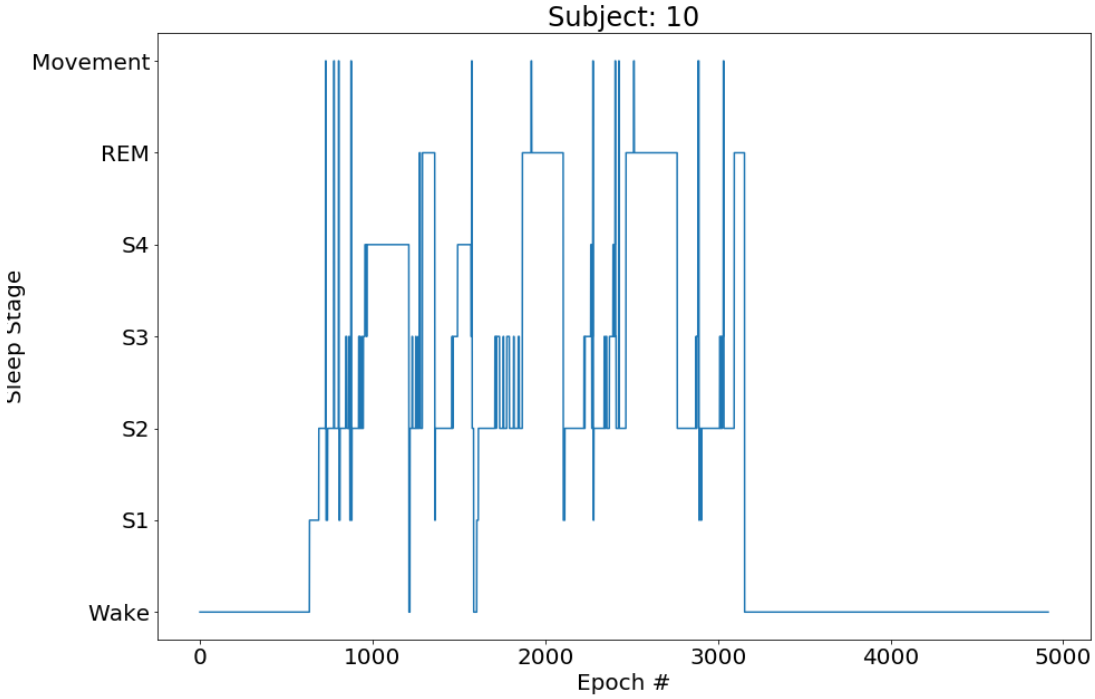

Subject: 5
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

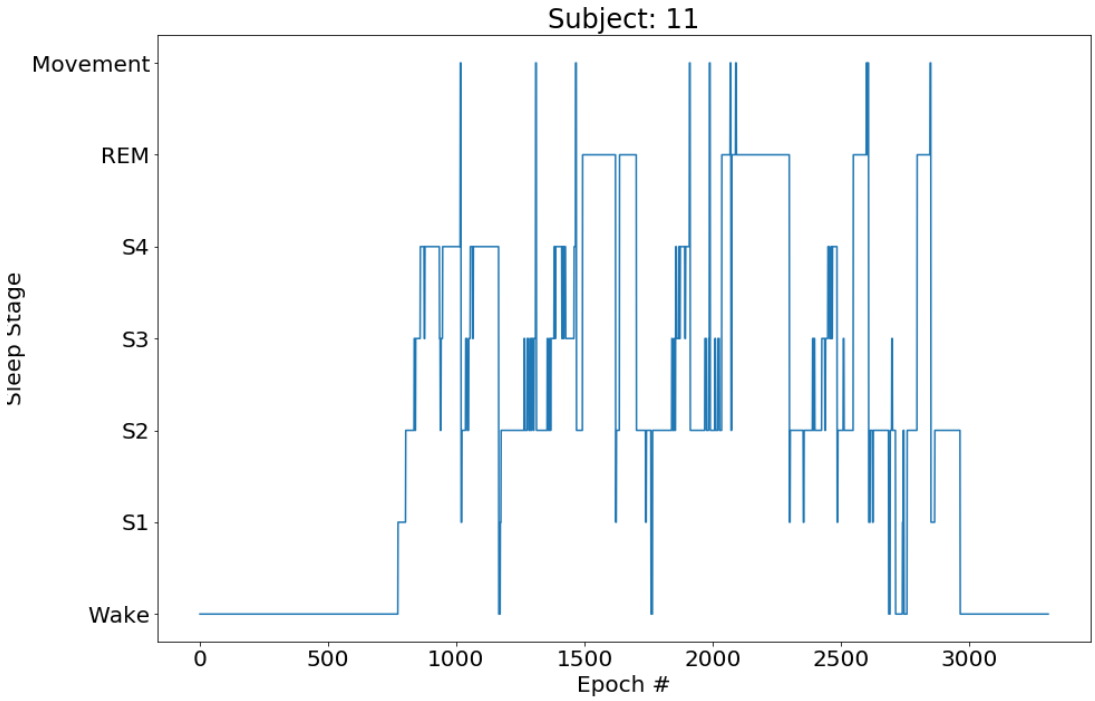

Subject: 6
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

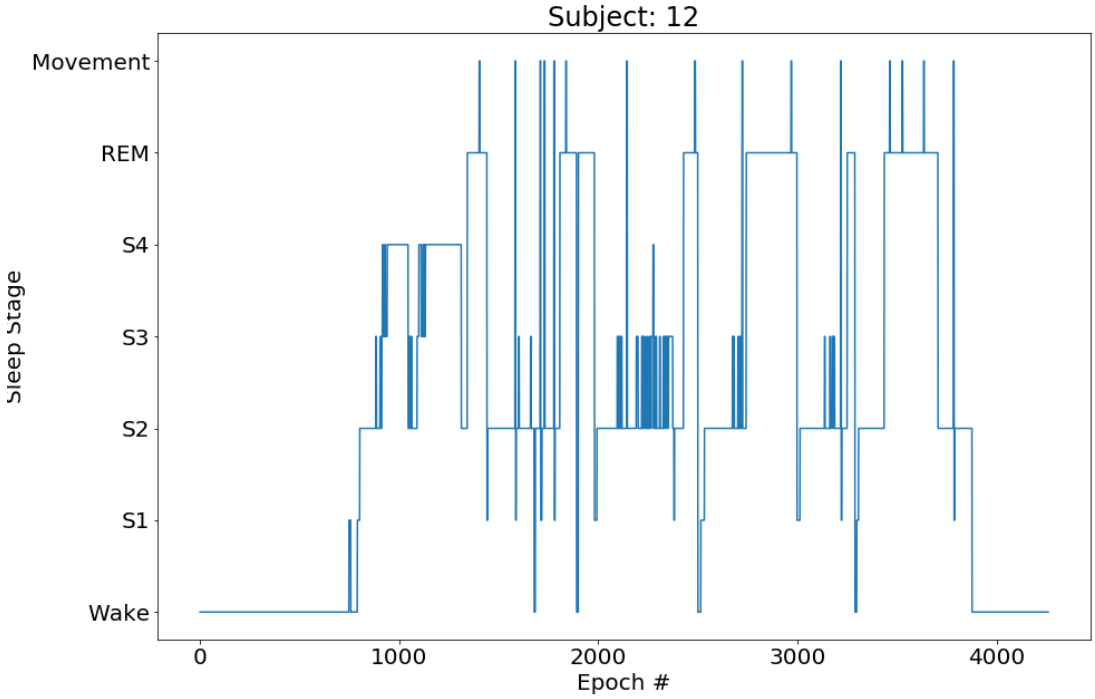

Subject: 7
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

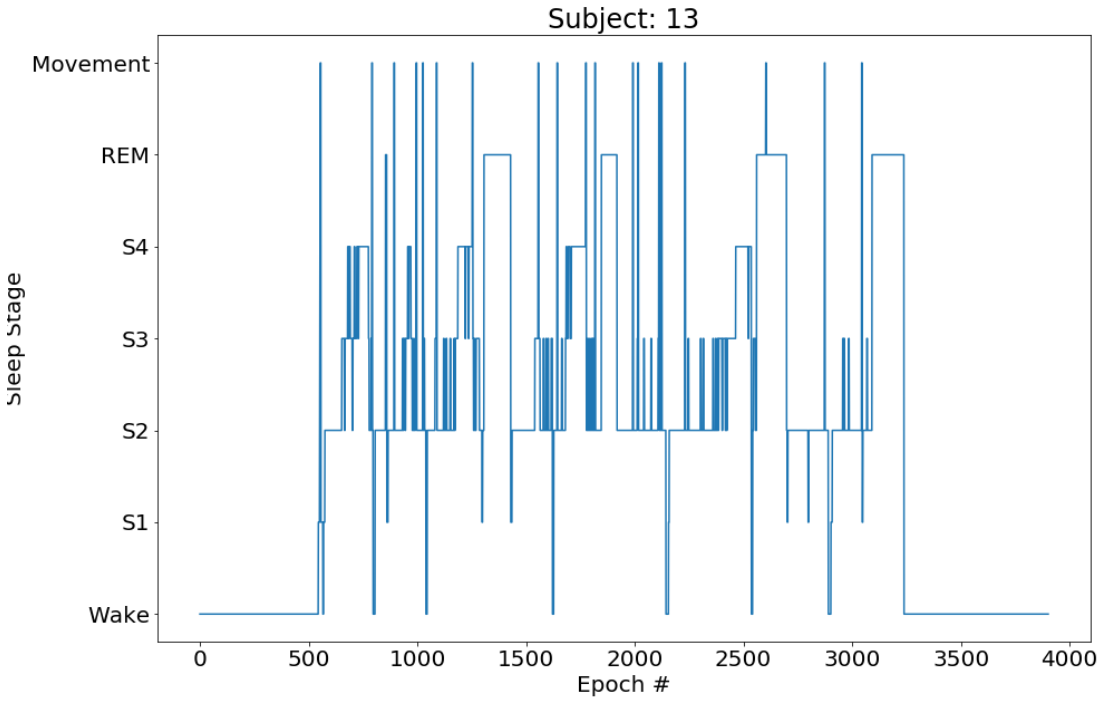

Subject: 8
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

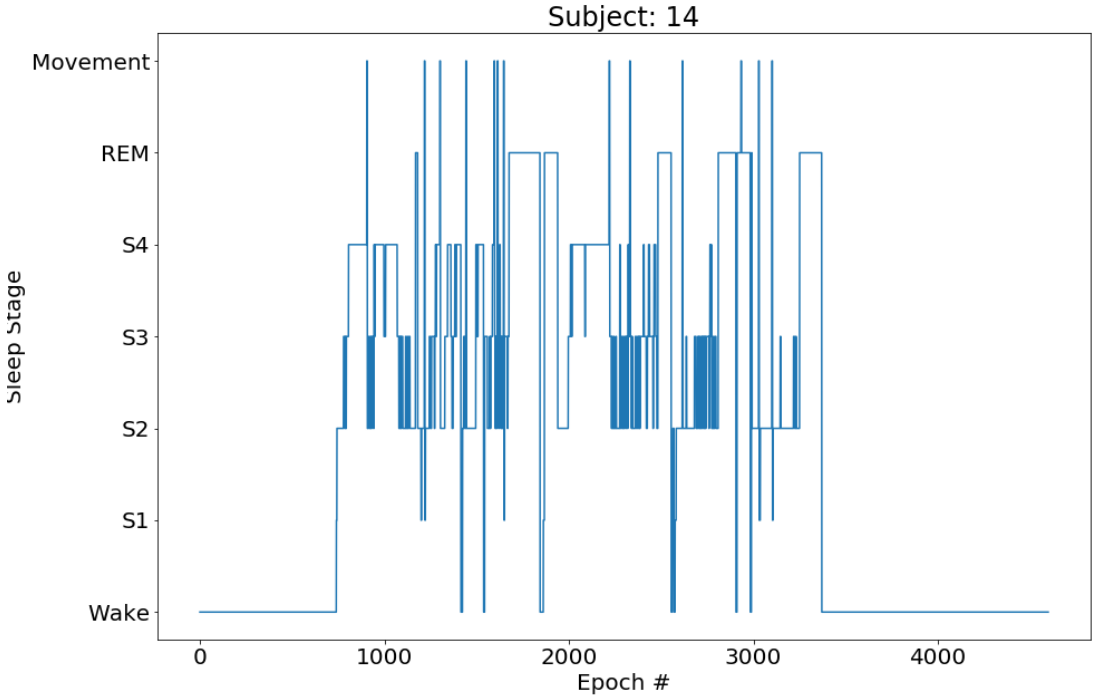

Subject: 9
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

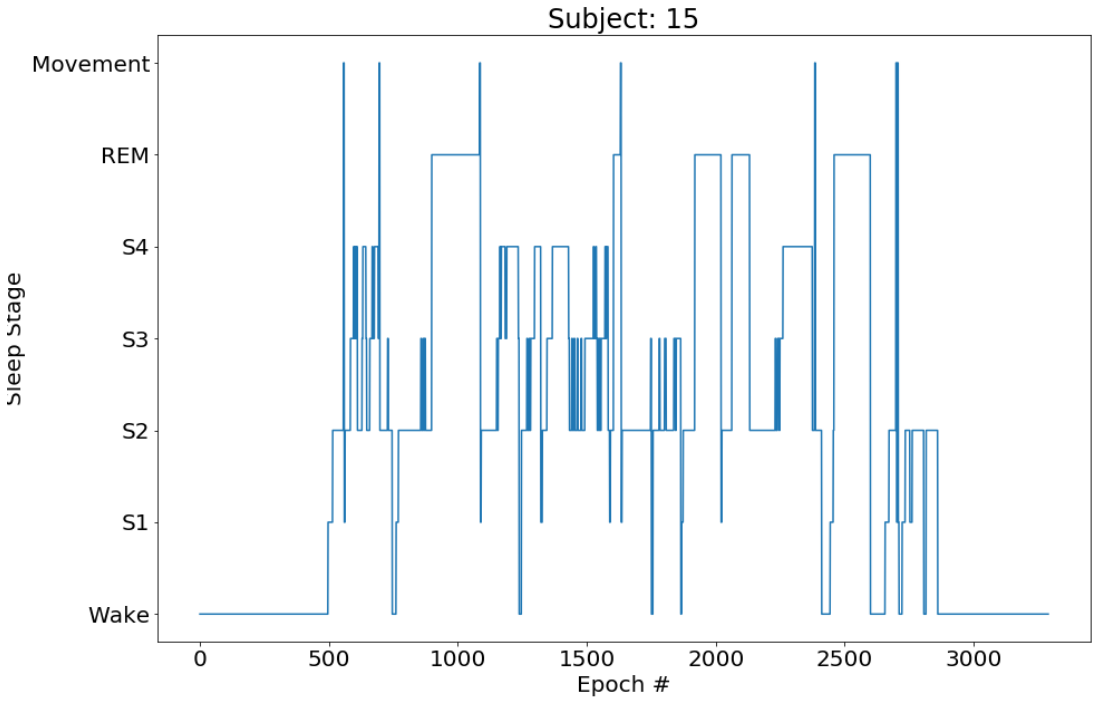

Subject: 10
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

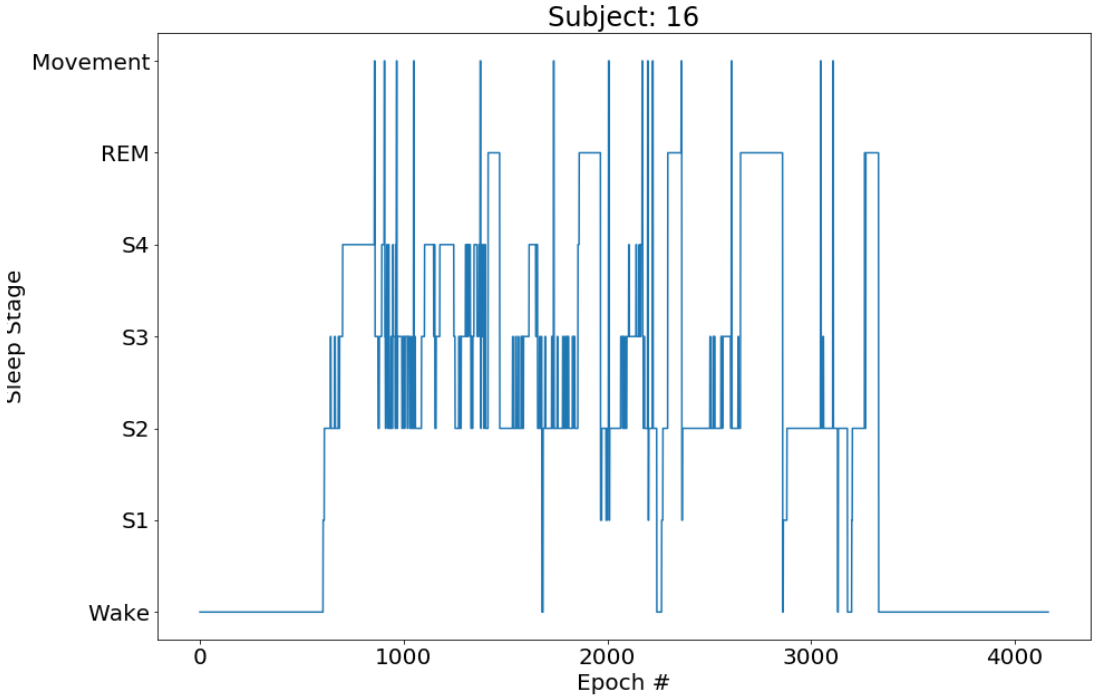

Subject: 11
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
Epoch #

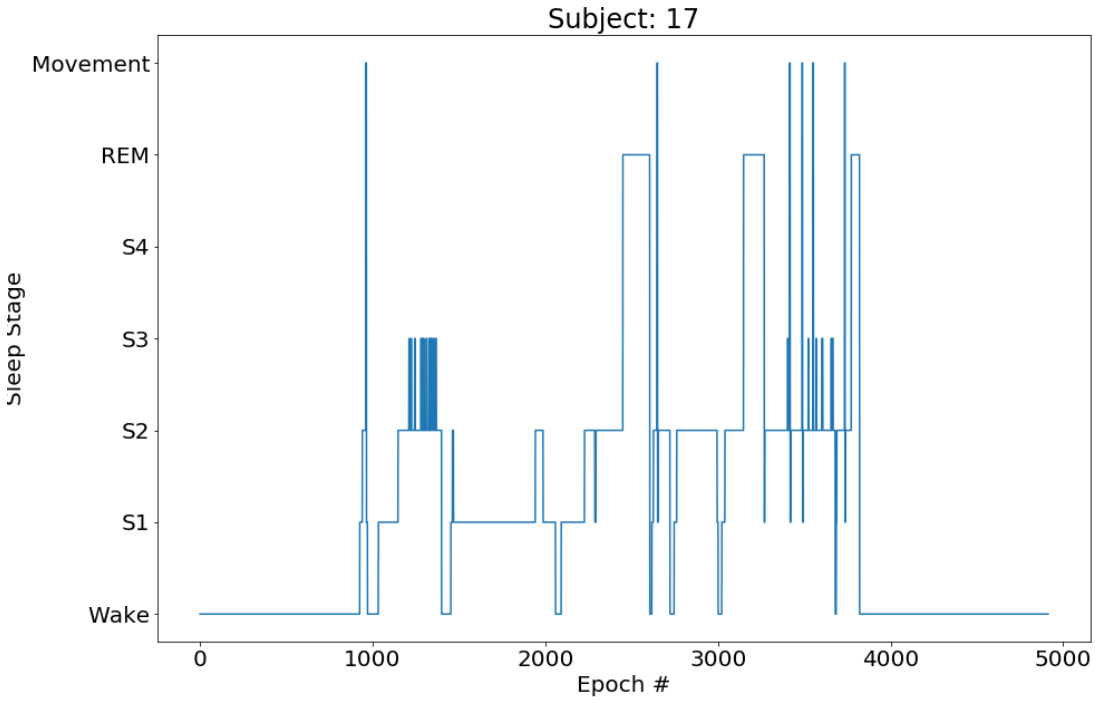

Subject: 12
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

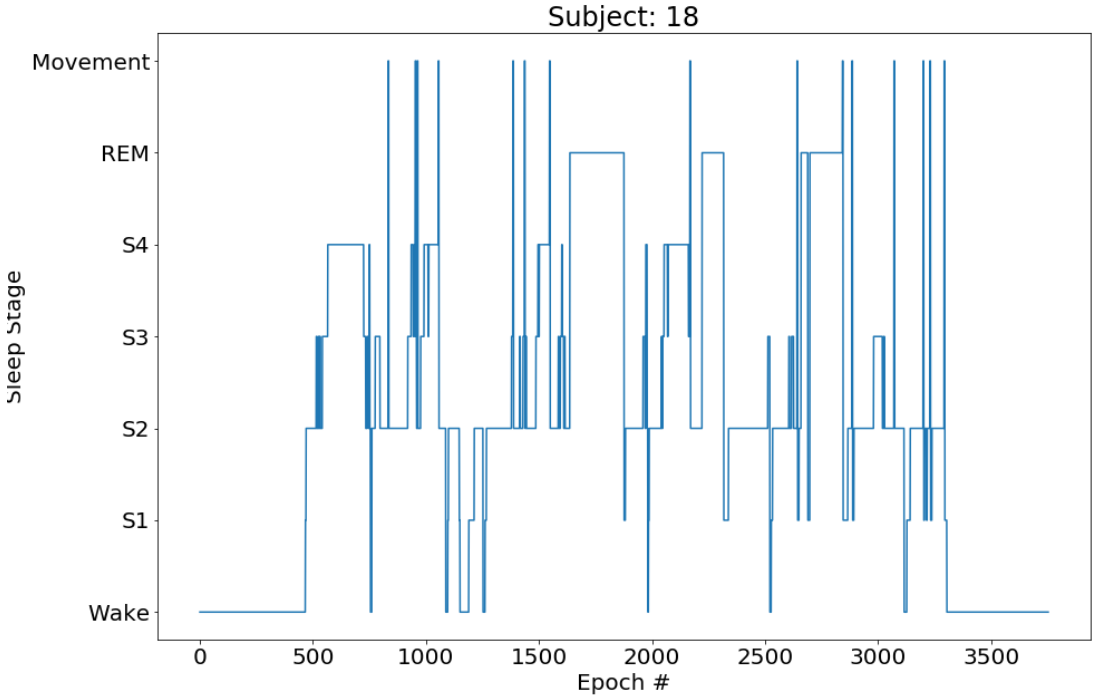

Subject: 13
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

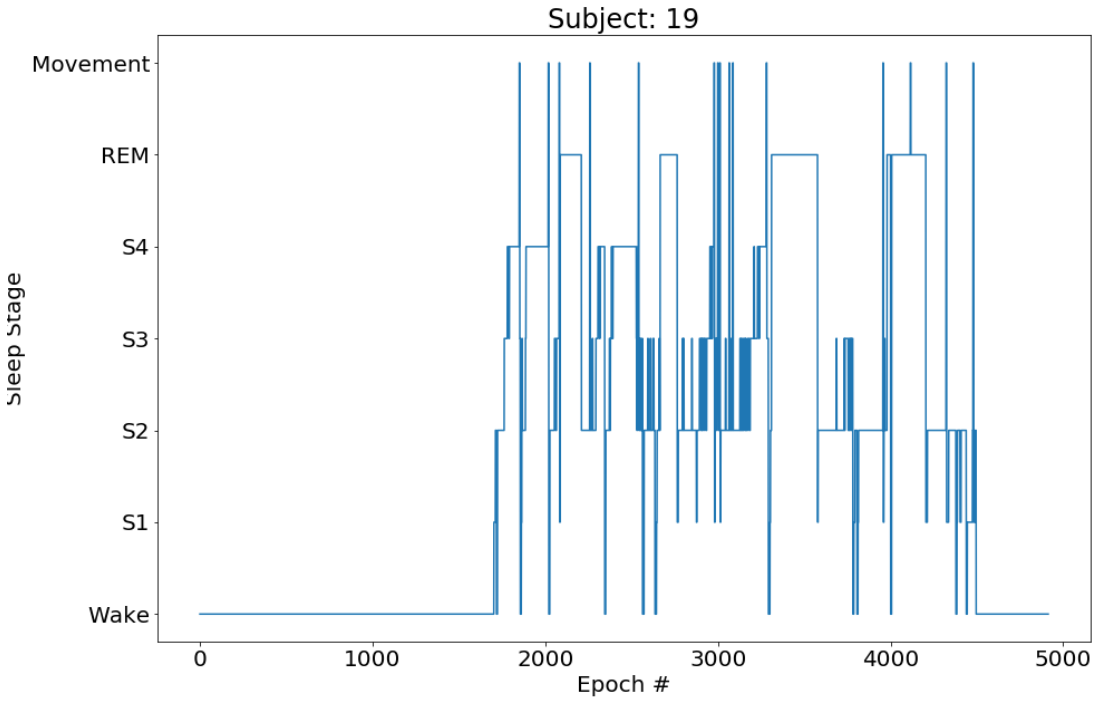

Subject: 14
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

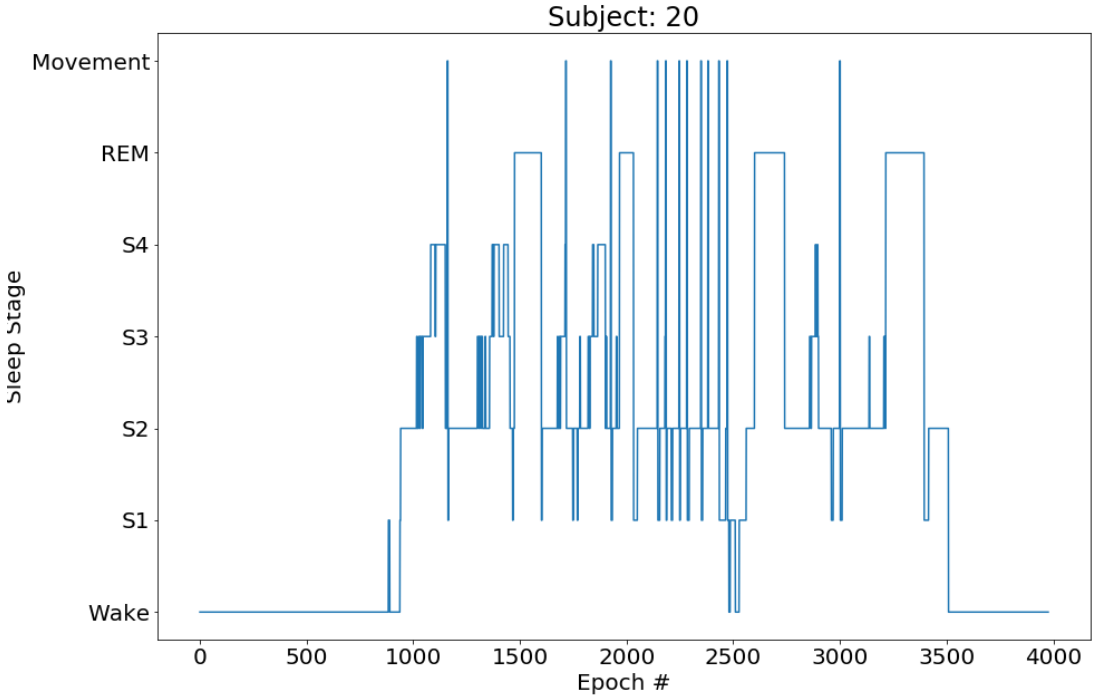

Subject: 15
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
Epoch #

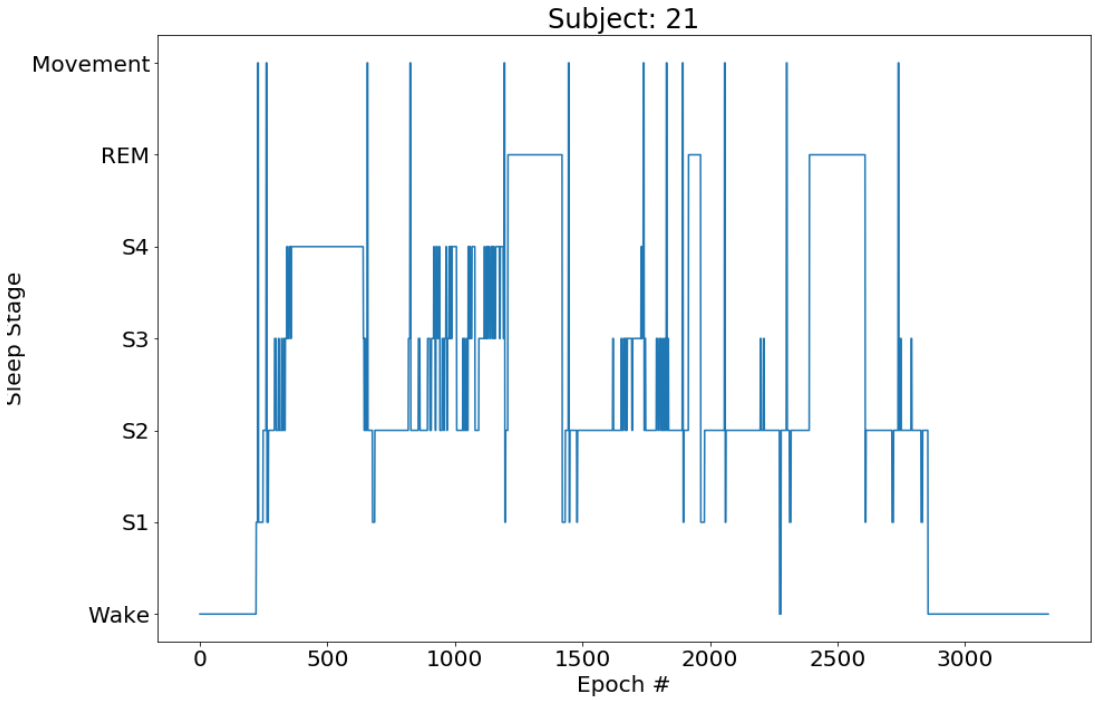

Subject: 16
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

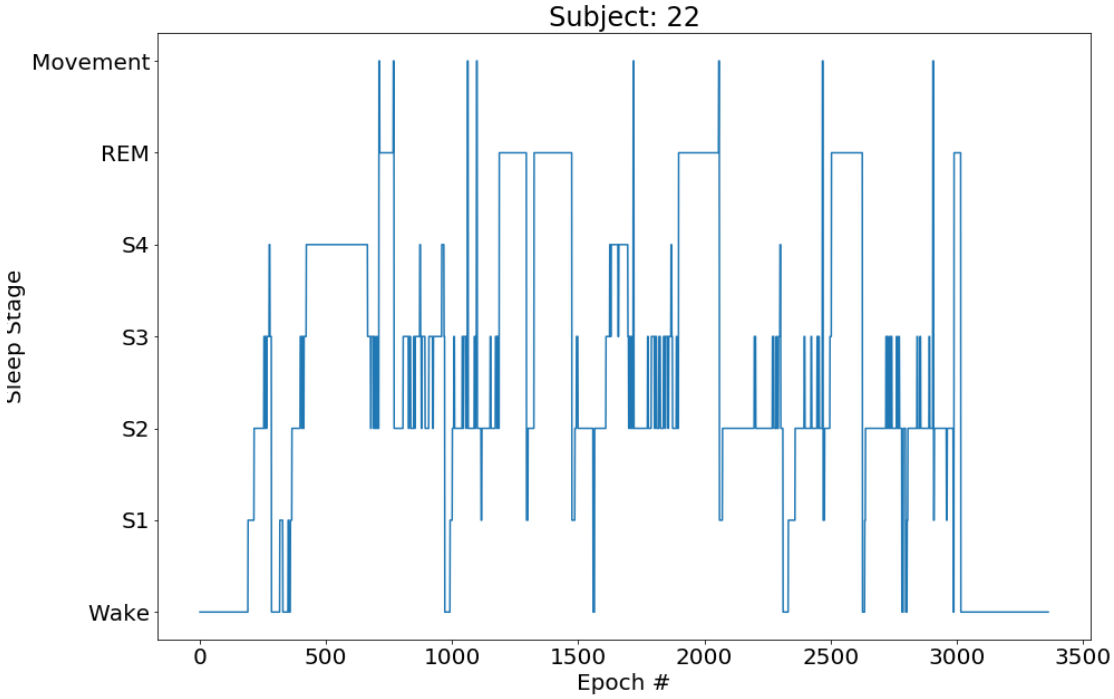

Subject: 17
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

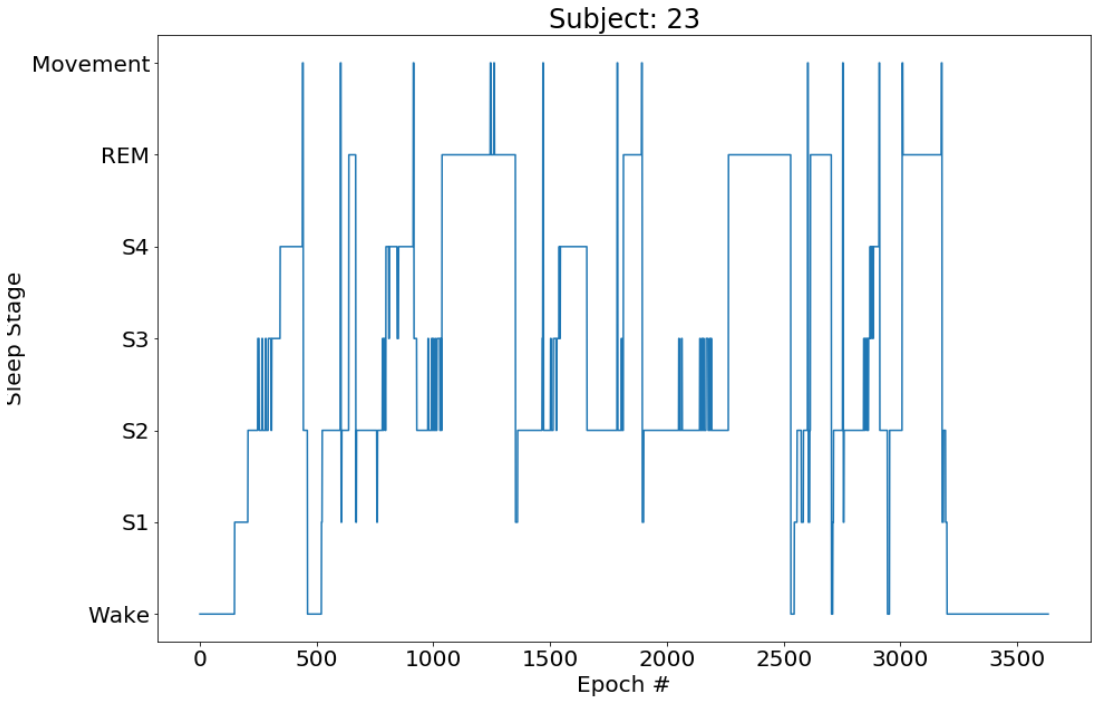

Subject: 18
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #

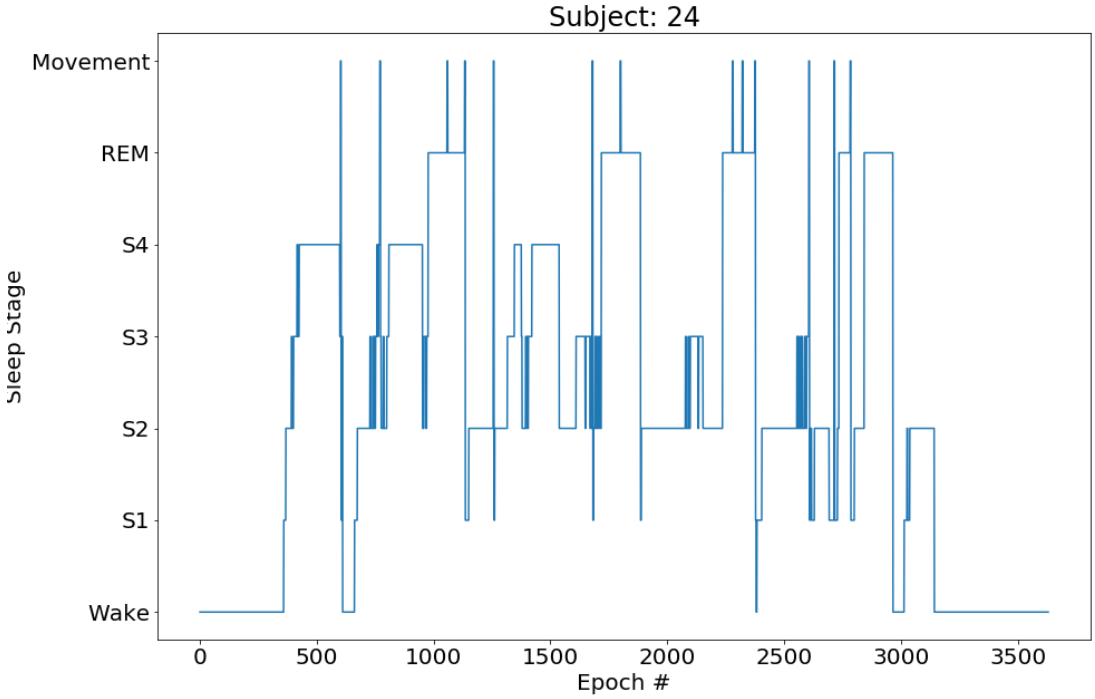

Subject: 19
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

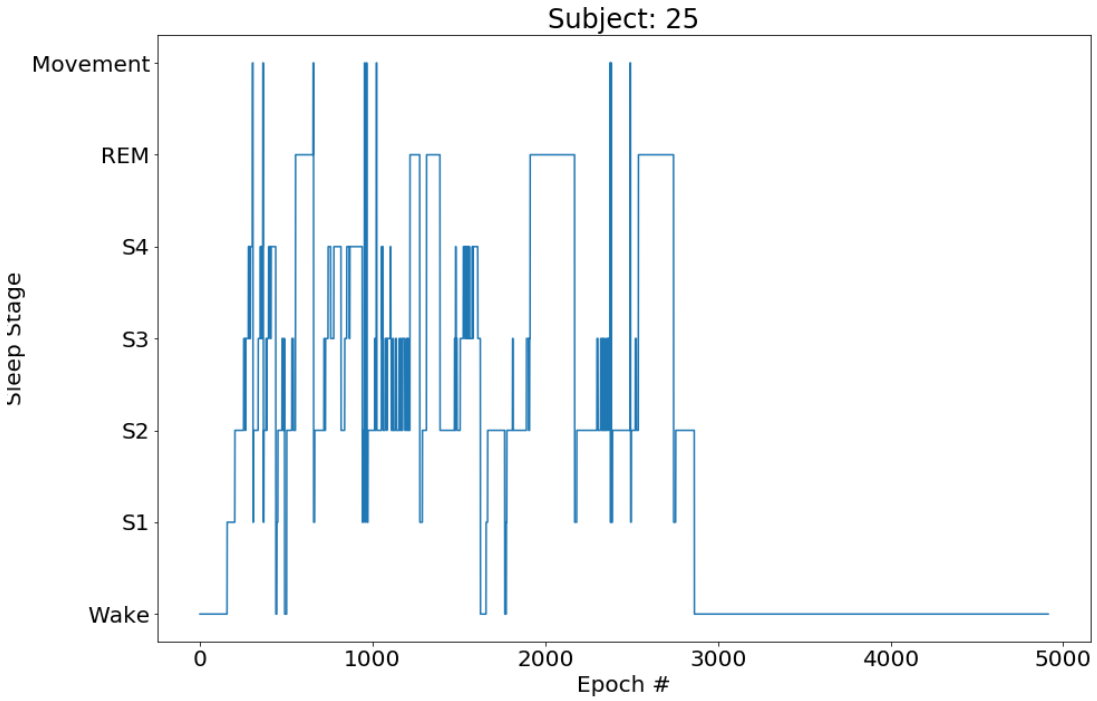

Subject: 20
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

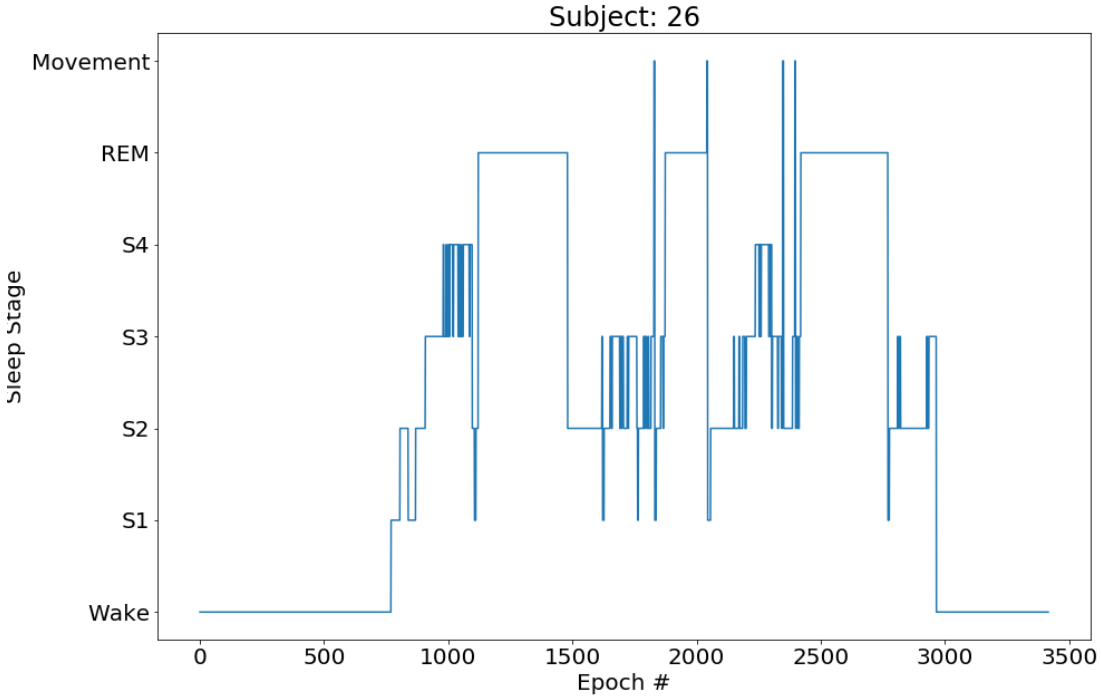

Subject: 21
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
Epoch #

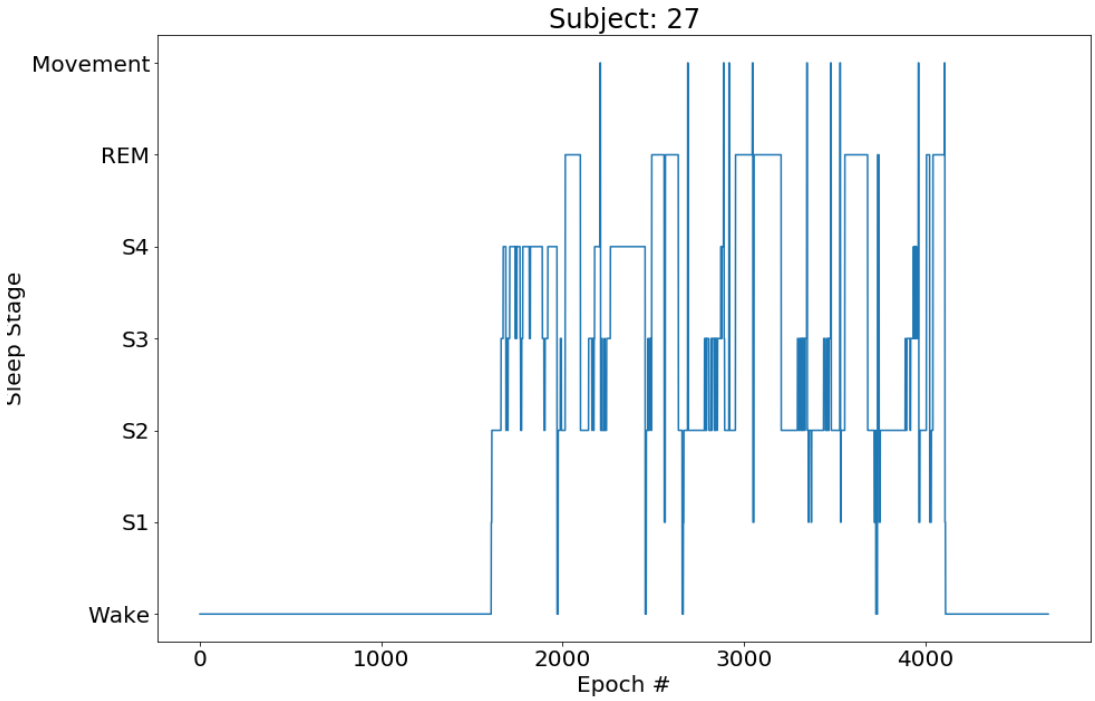

Subject: 22
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #

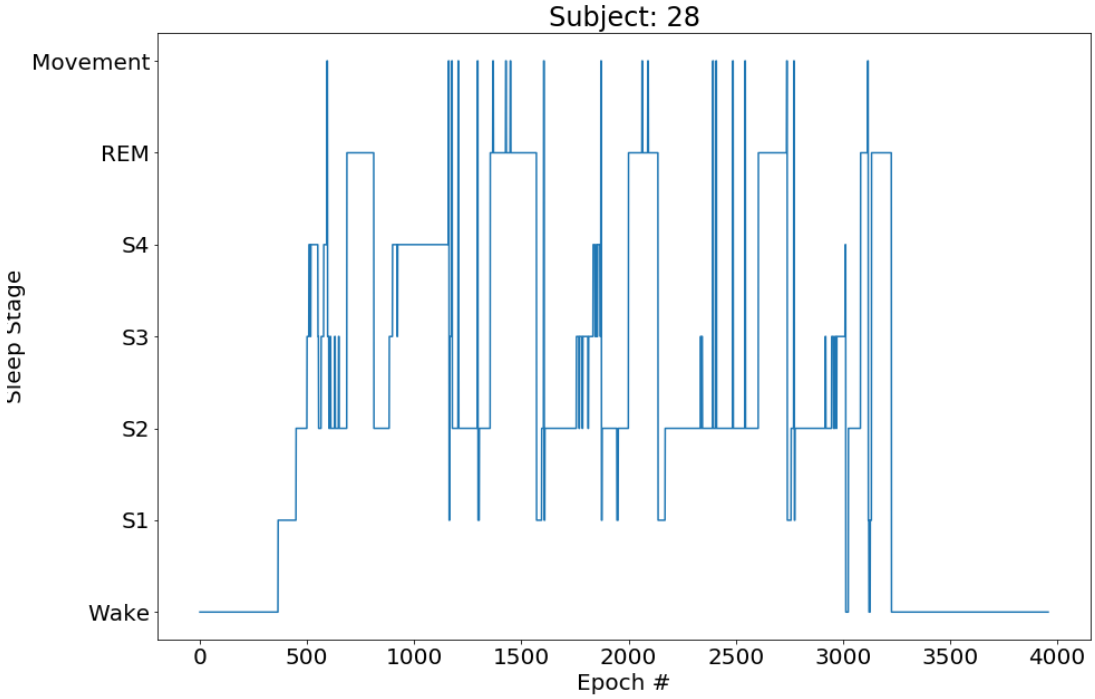

Subject: 23
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #

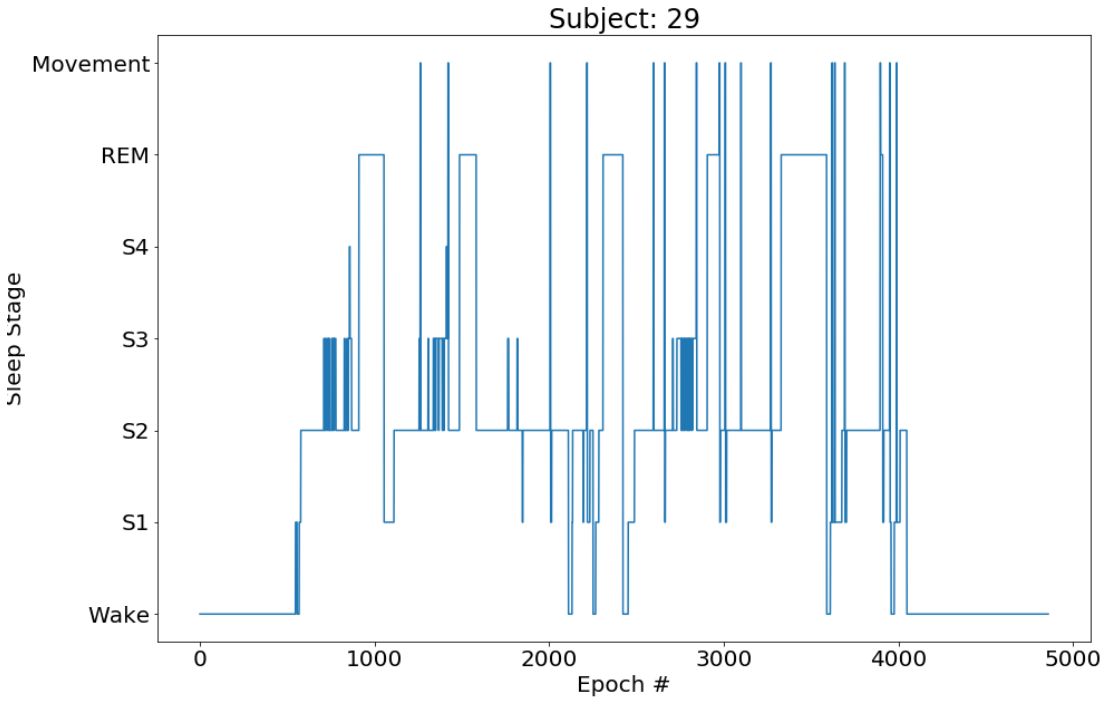

Subject: 24
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #

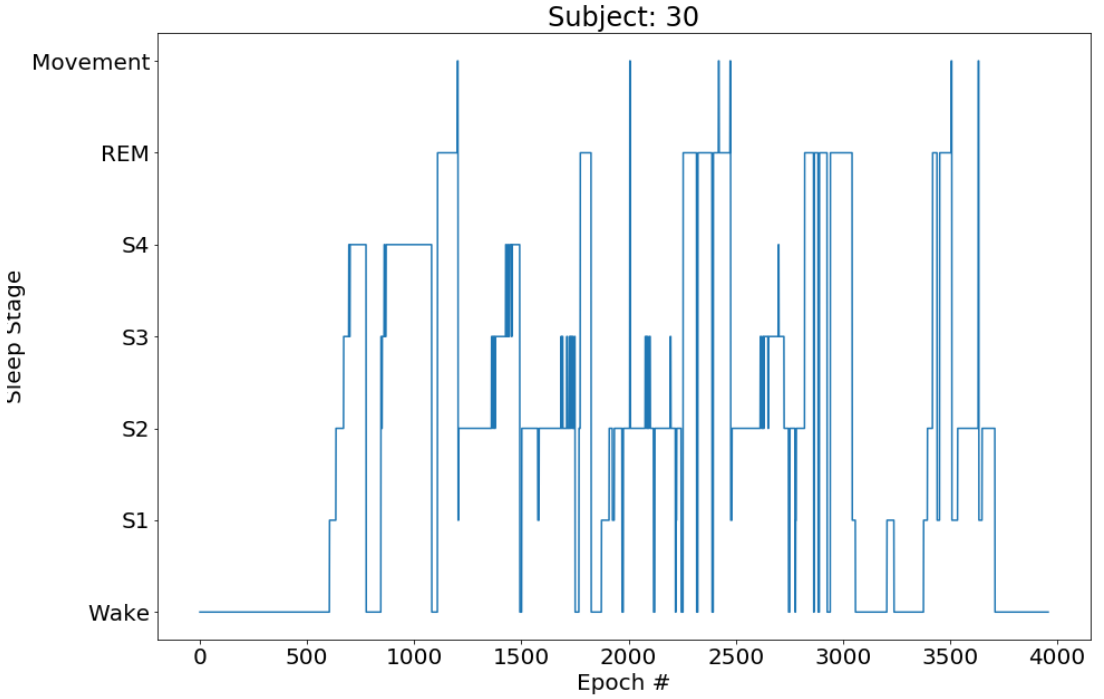

Subject: 25
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

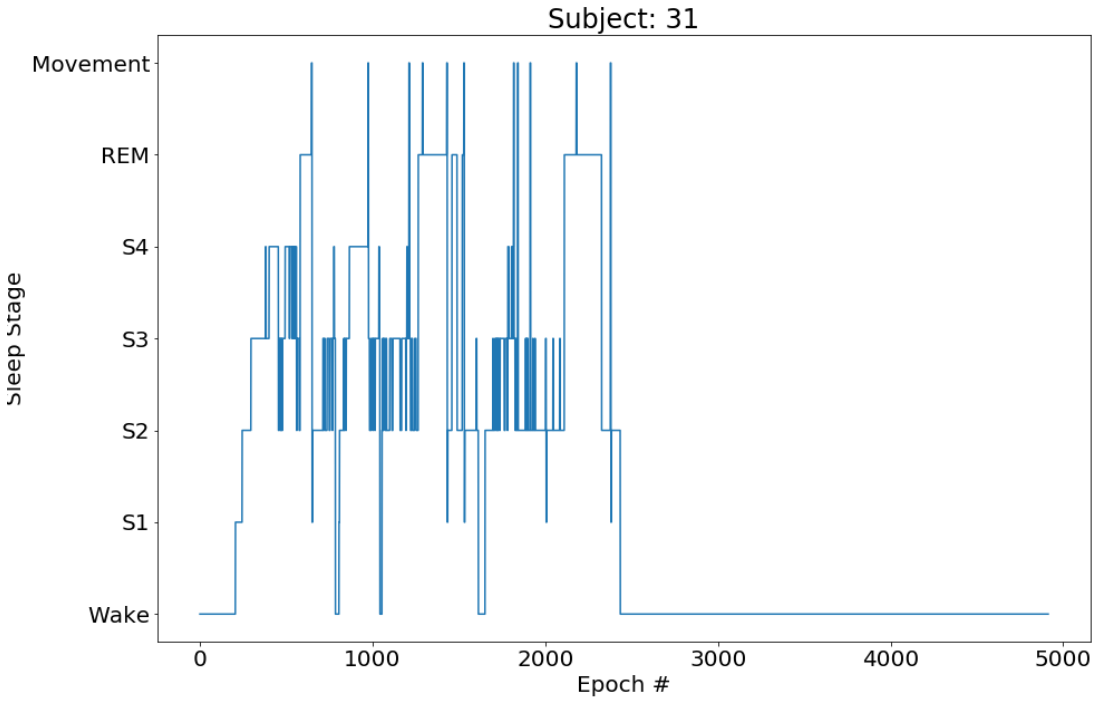

Subject: 26
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #

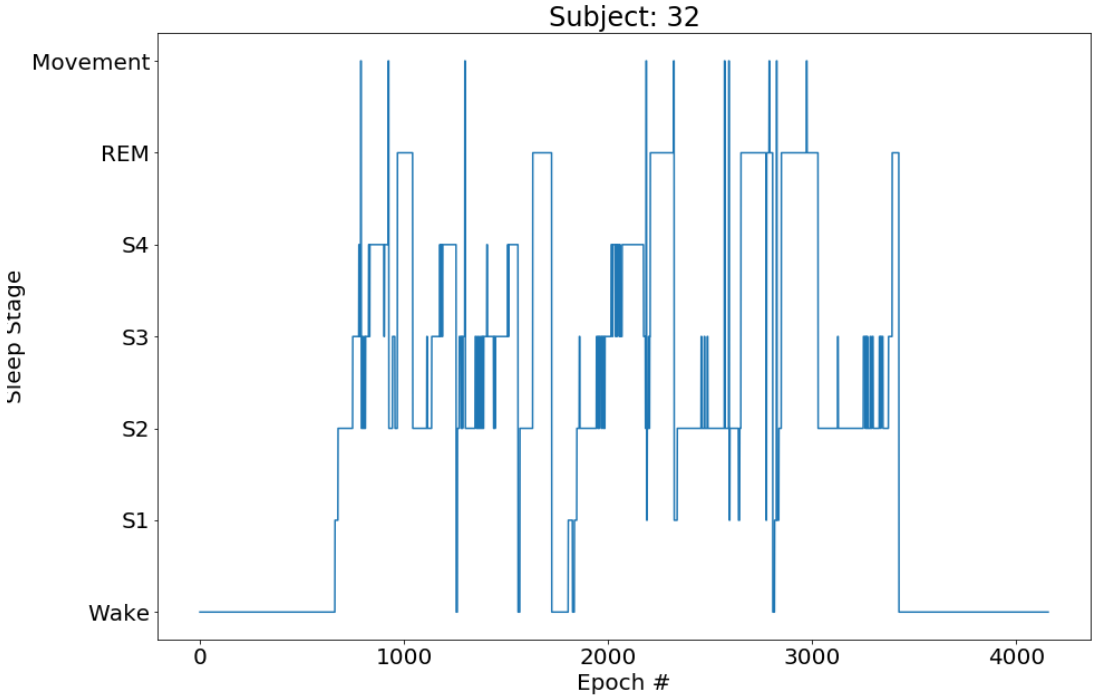

Subject: 27
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

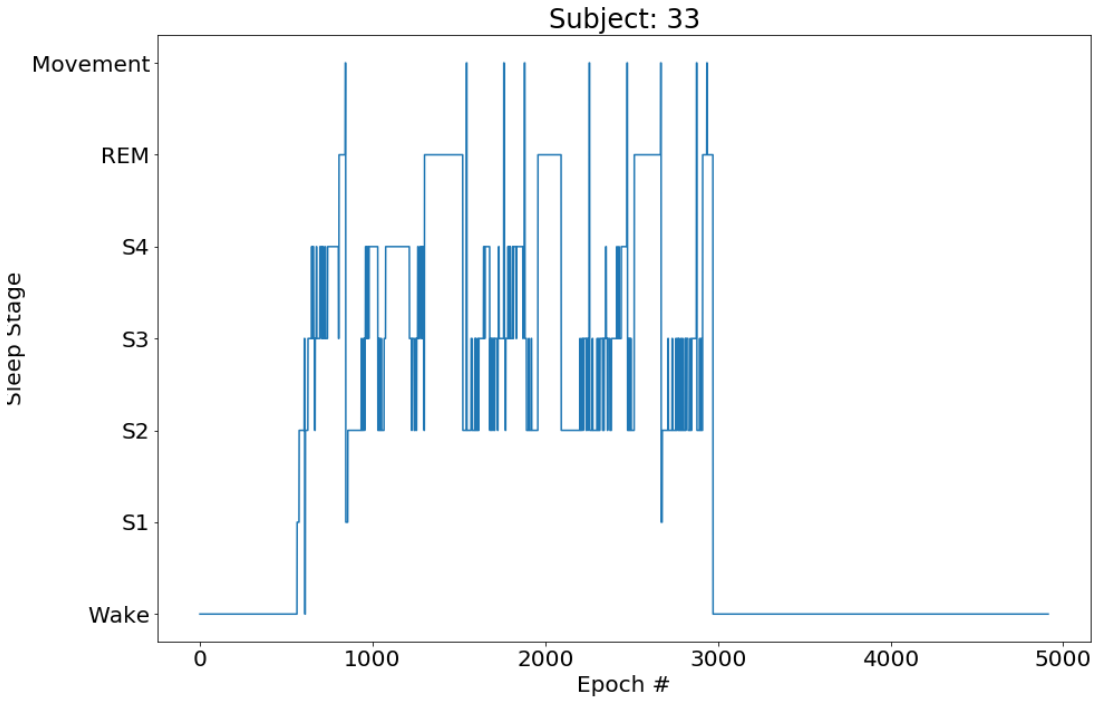

Subject: 28
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

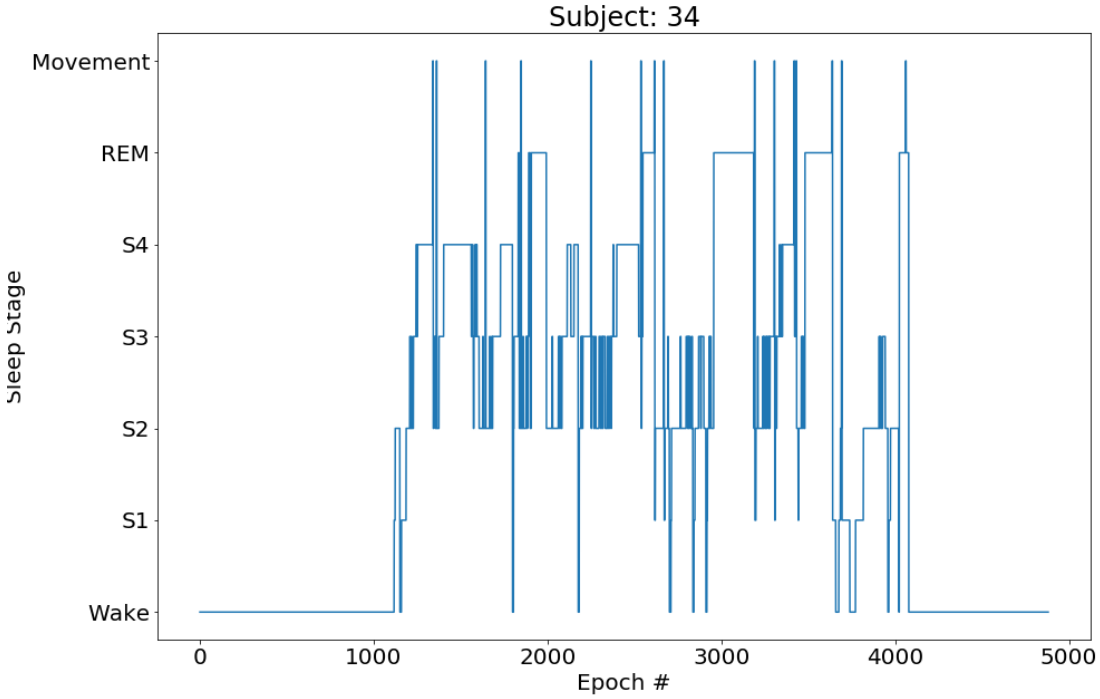

Subject: 29
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

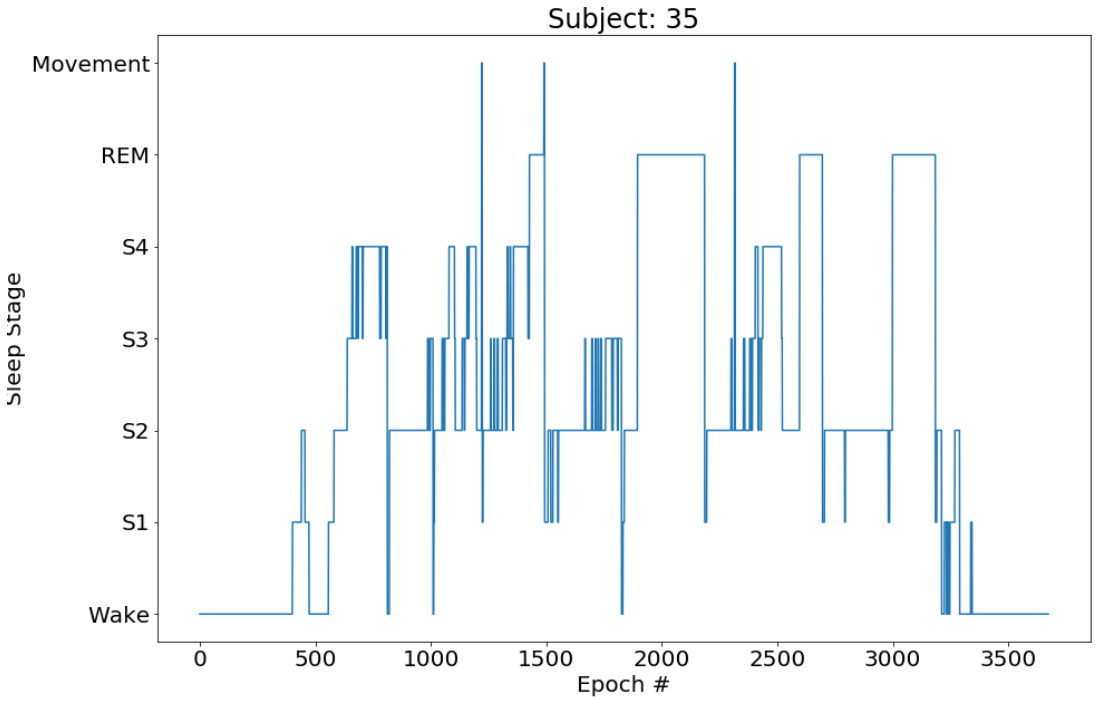

Subject: 30
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

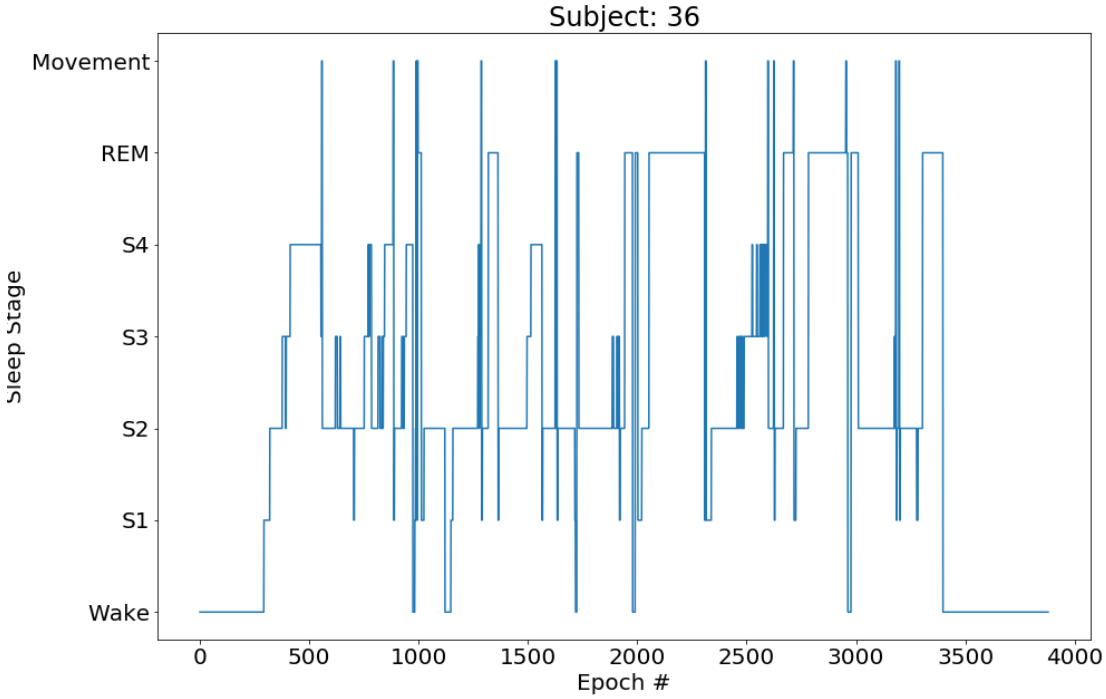

Subject: 31
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

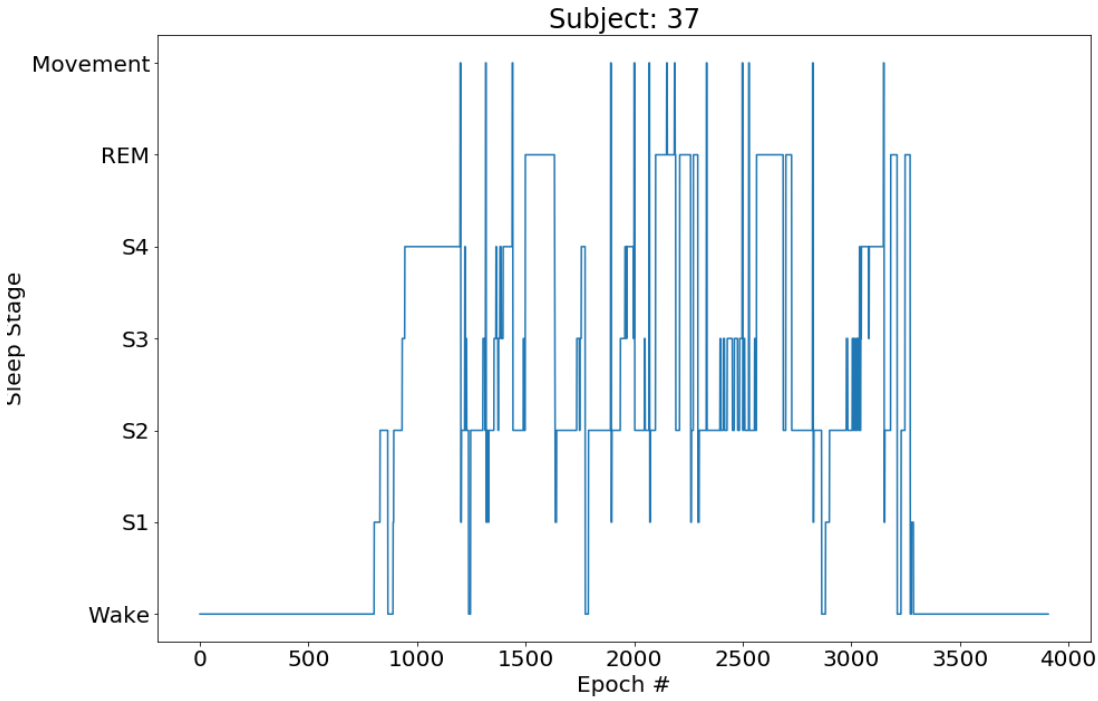

Subject: 32
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

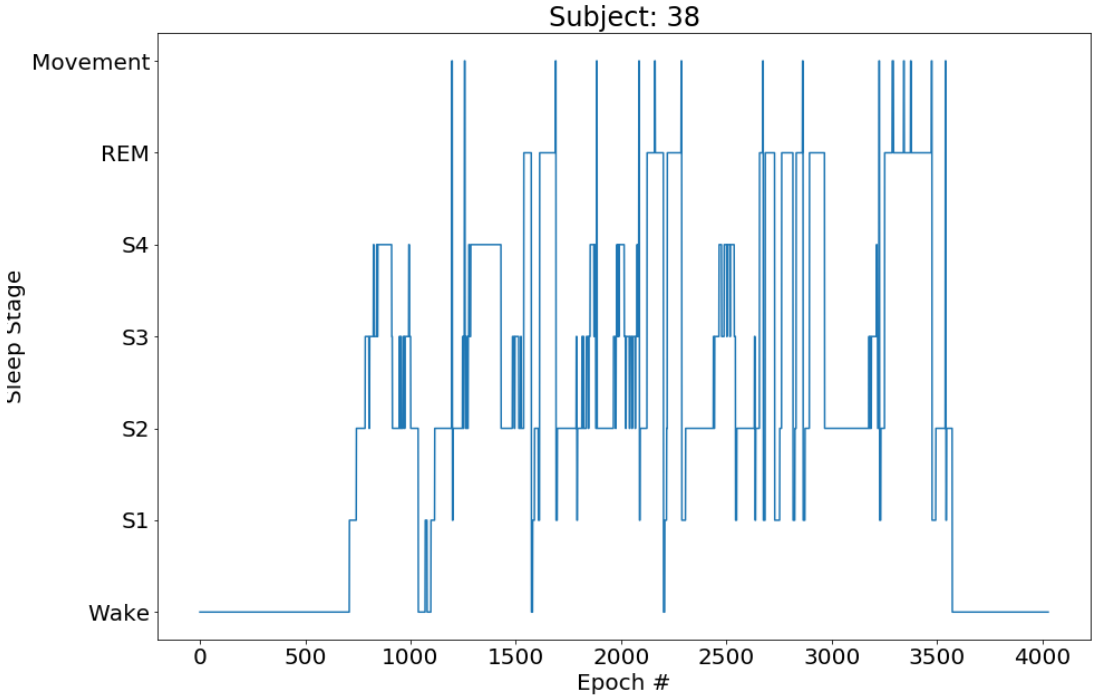

Subject: 33
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

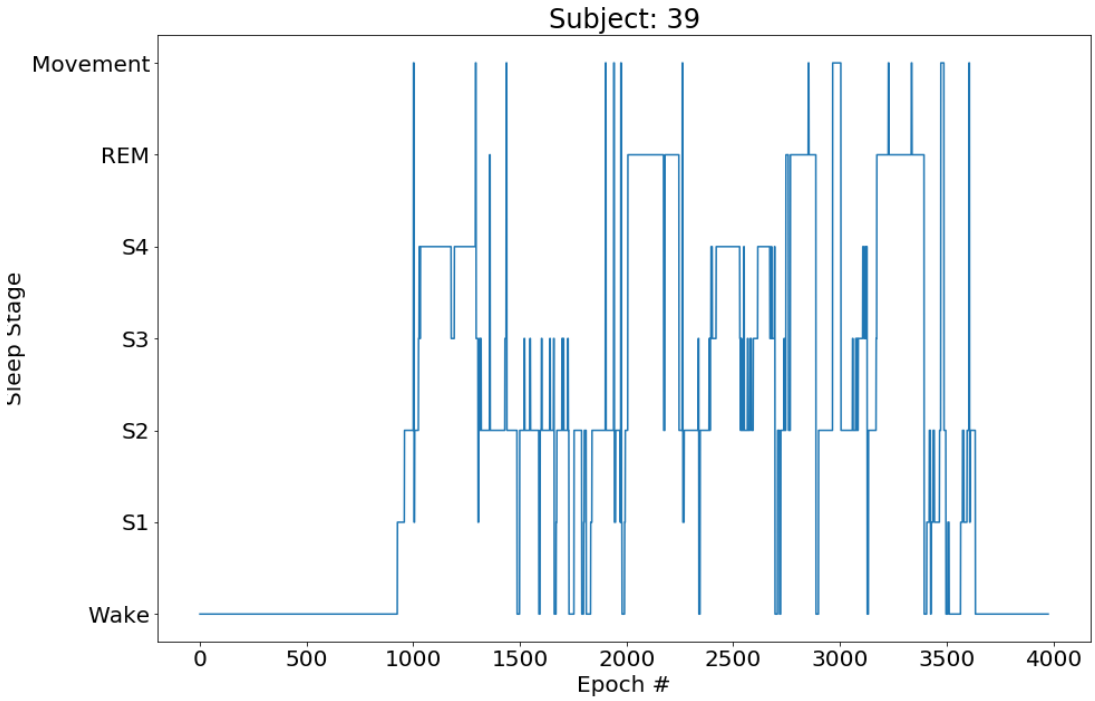

Subject: 34
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

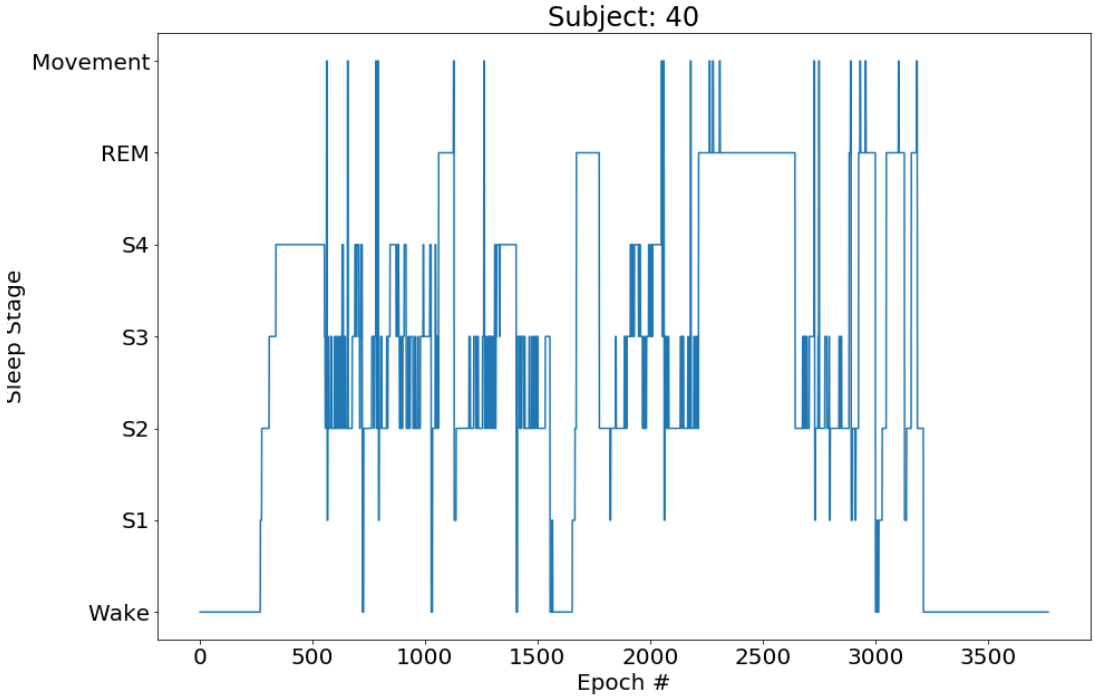

Subject: 35
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #

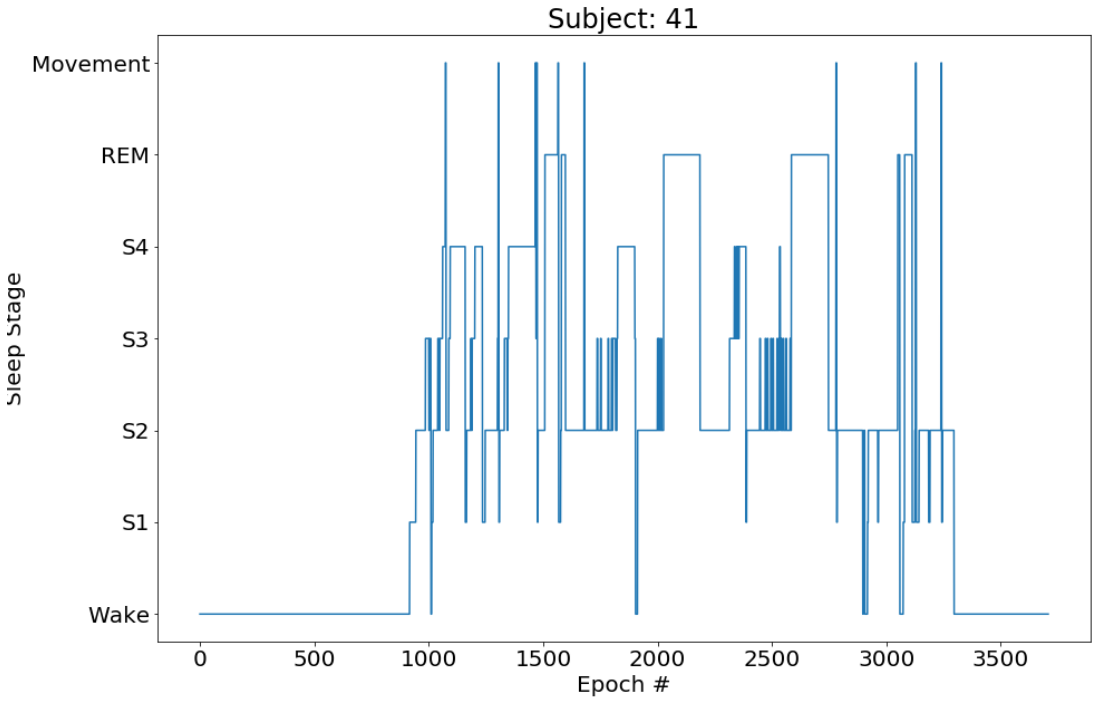

Subject: 36
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

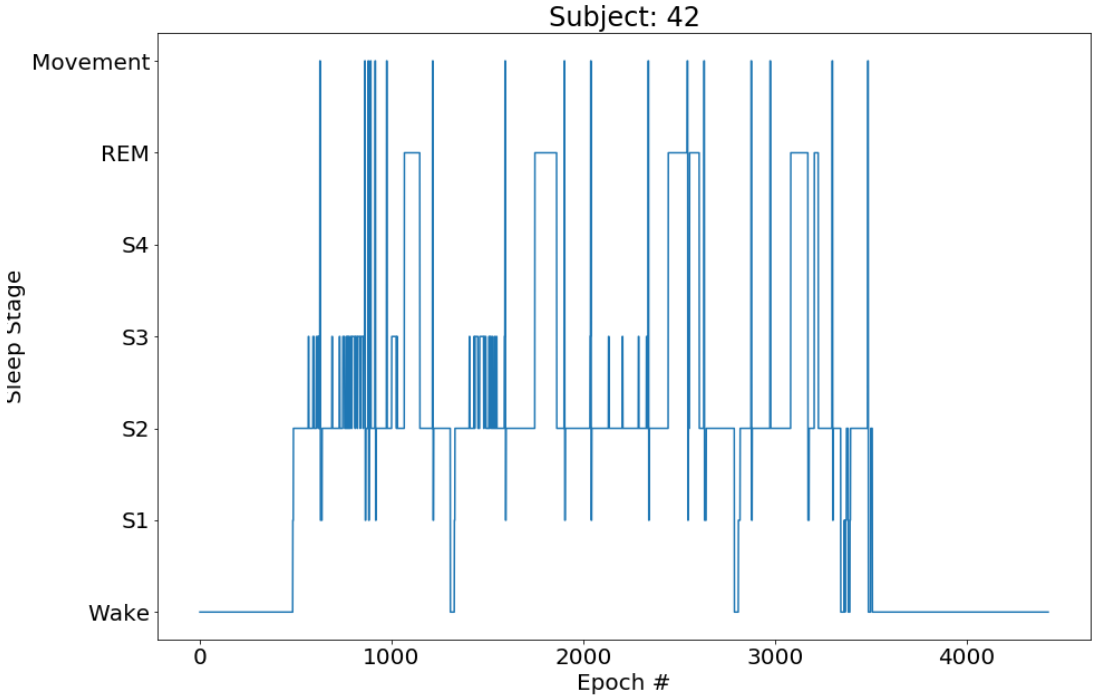

Subject: 37
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

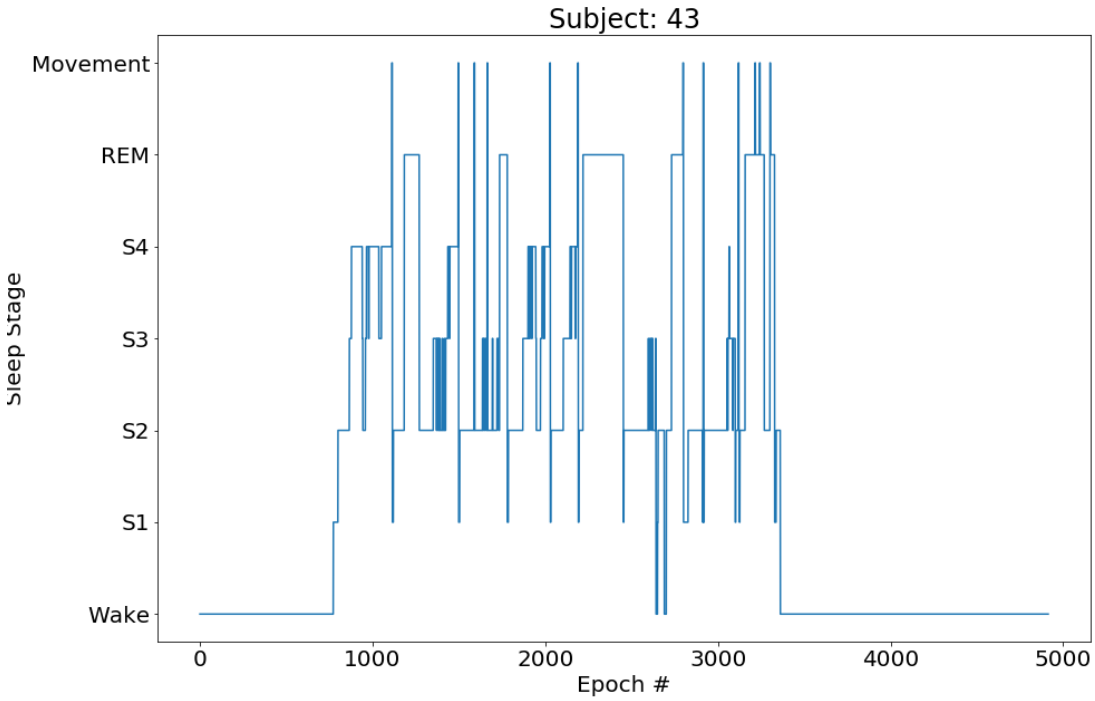

Subject: 38
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

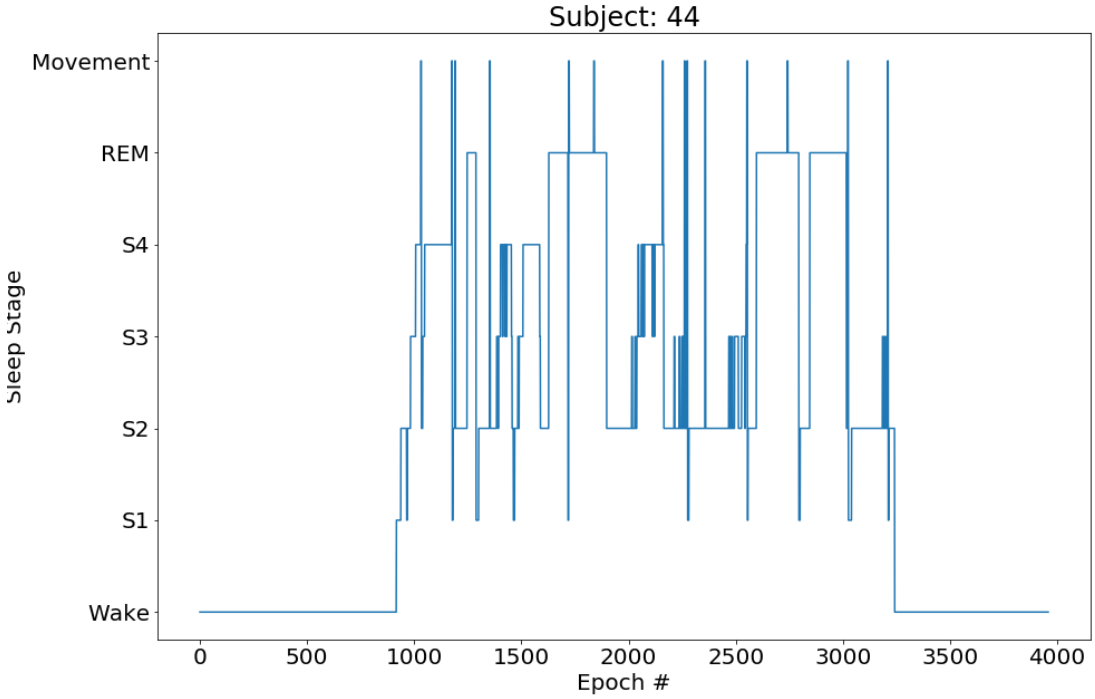

Subject: 39
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

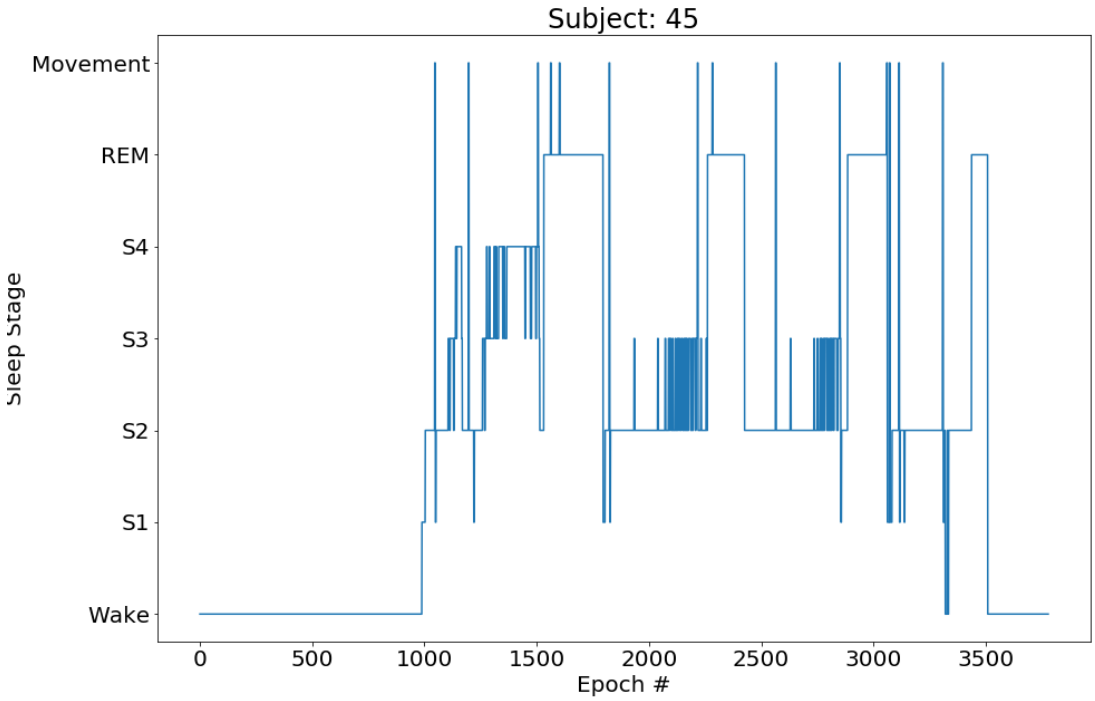

Subject: 40
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #

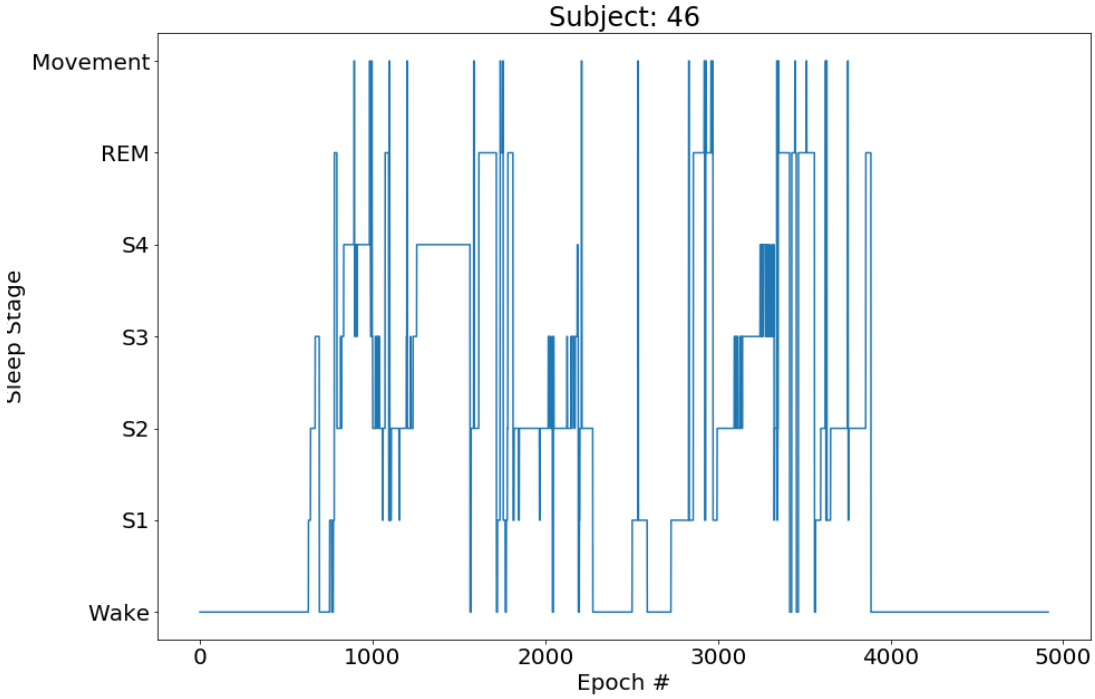

Subject: 41
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #

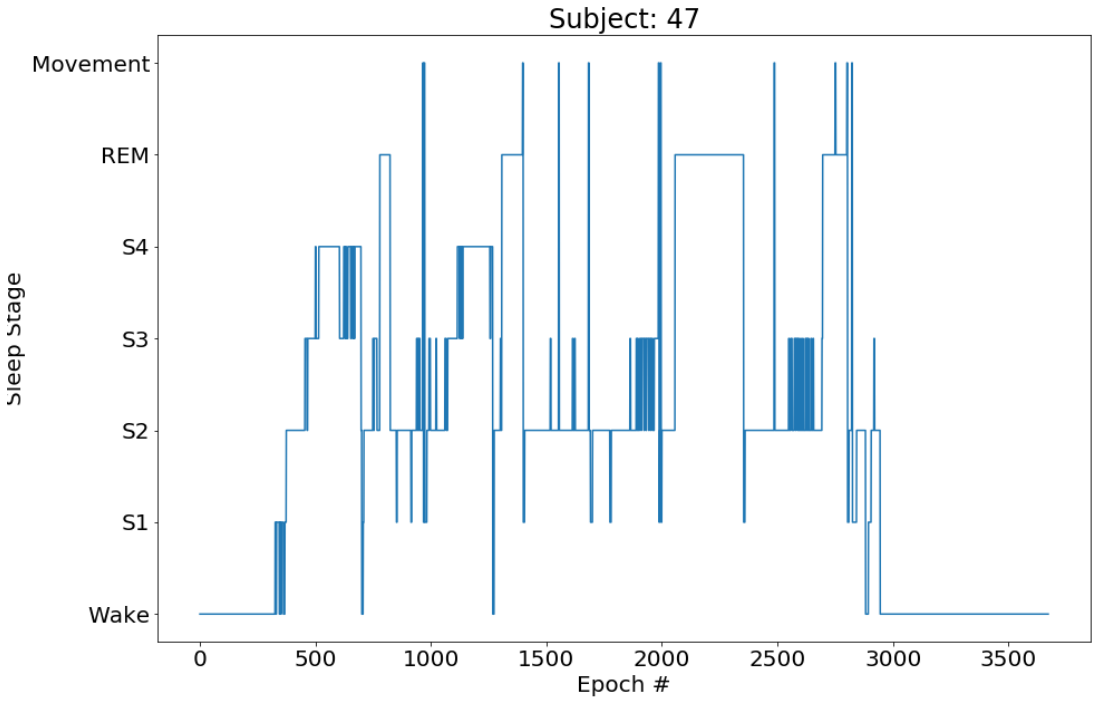

Subject: 42
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

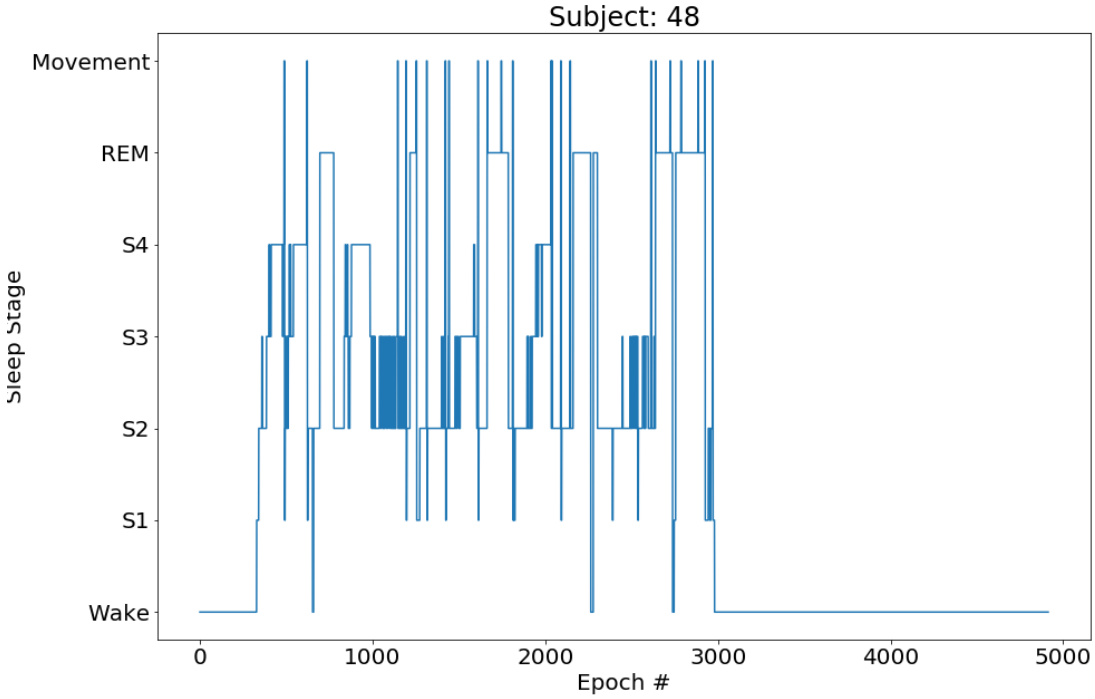

Subject: 43
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #

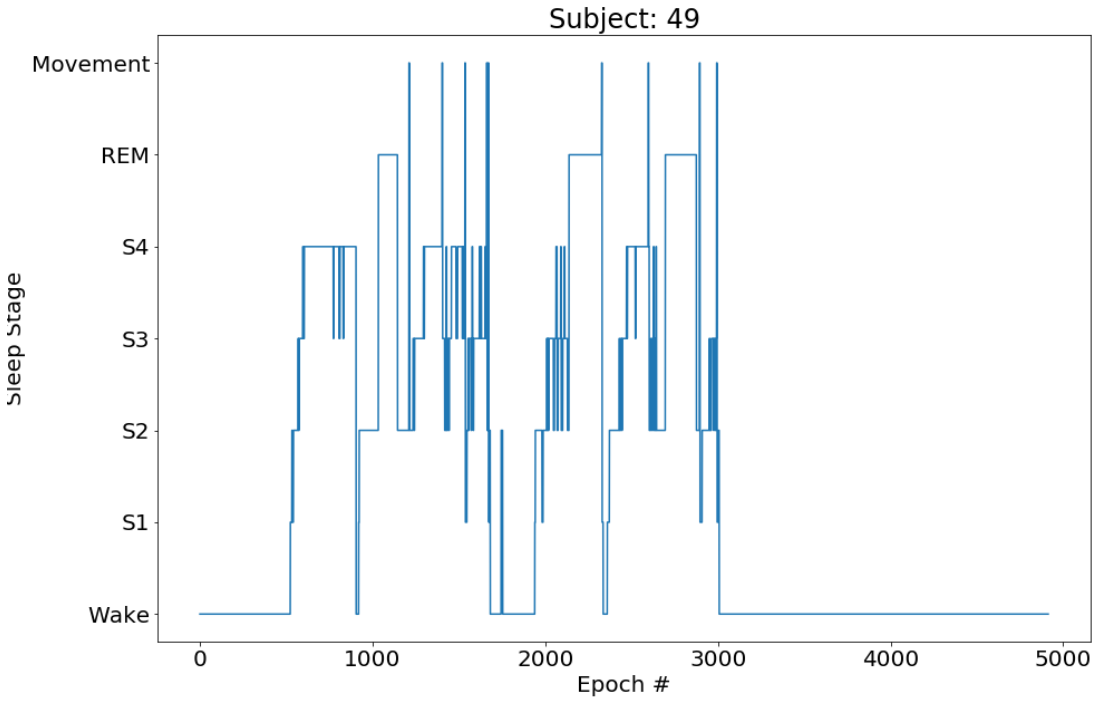

Subject: 44
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
4000
Epoch #

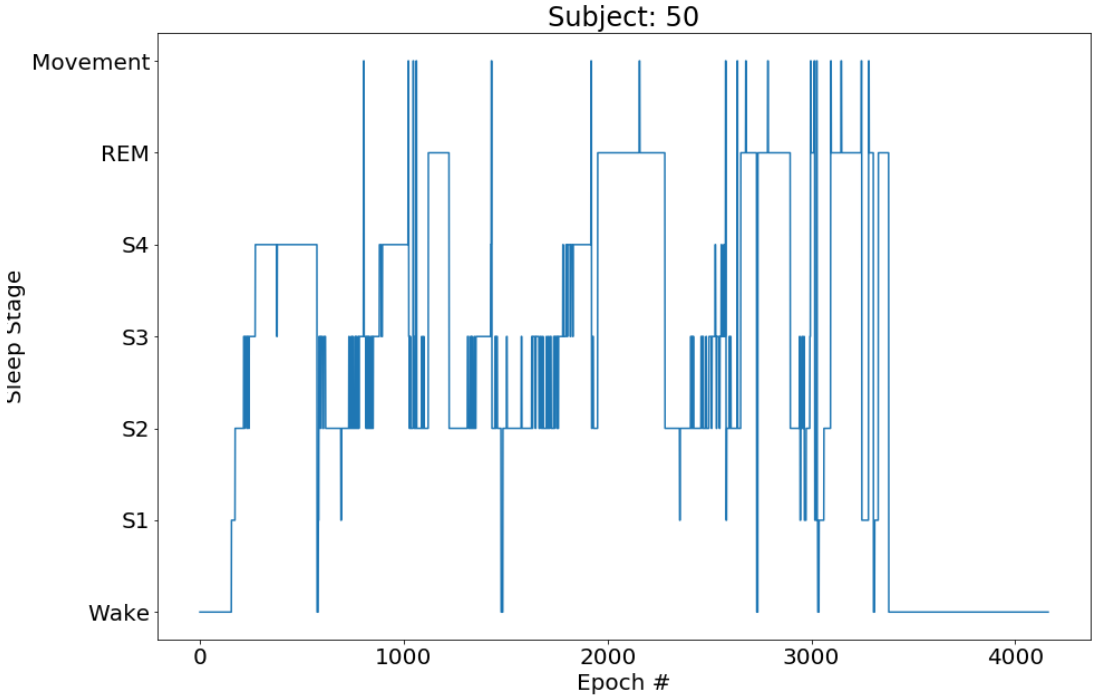

Subject: 45
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #


Subject: 46
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #


Subject: 47
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
500
1000
1500
2000
2500
3000
3500
Epoch #


Subject: 48
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #


Subject: 49
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
5000
Epoch #


Subject: 50
Movement
REM
S4
S3
S2
S1
Wake
Sleep Stage
0
1000
2000
3000
4000
Epoch #

## Appendix B

Time-domain features Sensitivity and Specificity table

| Channel | | Sensitivity | Specificity |
|---|---|---|---|
| Oz-Cz | RF 3 class | 0.9, 0.9, 0.4 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.7, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.8, 0.3 | 0.8, 0.8, 1.0, |
| | KNN 7 class | 0.9, 0.0, 0.7, 0.1, 0.7, 0.3 | 0.8, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Oz-Fpz | RF 3 class | 0.9, 0.9, 0.4 | 0.9, 0.8, 1.0, |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.7, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.9, 0.8, 0.2 | 0.9, 0.7, 1.0, |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.7, 0.3 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Oz-C3 | RF 3 class | 0.9, 0.9, 0.4 | 0.9, 0.8, 1.0, |
| | RF 7 class | 0.9, 0.1, 0.7, 0.3, 0.7, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.9, 0.8, 0.3 | 0.9, 0.8, 1.0, |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.7, 0.4 | 0.8, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Oz-C4 | RF 3 class | 0.9, 0.9, 0.4 | 0.9, 0.8, 1.0, |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.7, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.9, 0.8, 0.3 | 0.8, 0.8, 1.0, |
| | KNN 7 class | 0.9, 0.0, 0.7, 0.1, 0.7, 0.4 | 0.8, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Cz-Fpz | RF 3 class | 0.8, 0.8, 0.4 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.8, 0.3 | 0.9, 0.7, 1.0, |
| | KNN 7 class | 0.8, 0.1, 0.7, 0.1, 0.7, 0.3 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Cz-C3 | RF 3 class | 0.8, 0.8, 0.3 | 0.9, 0.8, 1.0, |
| | RF 7 class | 0.8, 0.1, 0.7, 0.2, 0.6, 0.4 | 0.8, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.7, 0.8, 0.2 | 0.8, 0.7, 1.0, |
| | KNN 7 class | 0.8, 0.1, 0.6, 0.1, 0.6, 0.3 | 0.8, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Cz-C4 | RF 3 class | 0.8, 0.8, 0.3 | 0.9, 0.7, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.6, 0.2, 0.6, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.8, 0.2 | 0.8, 0.7, 1.0, |
| | KNN 7 class | 0.8, 0.0, 0.6, 0.0, 0.6, 0.3 | 0.7, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Fpz-C3 | RF 3 class | 0.9, 0.8, 0.4 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.7, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.9, 0.3 | 0.9, 0.7, 1.0, |
| | KNN 7 class | 0.8, 0.1, 0.7, 0.1, 0.7, 0.3 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Fpz-C4 | RF 3 class | 0.9, 0.8, 0.4 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.7, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.8, 0.2 | 0.9, 0.7, 1.0, |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.7, 0.3 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| C3-C4 | RF 3 class | 0.8, 0.8, 0.4 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.7, 0.5 | 0.8, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.8, 0.3 | 0.8, 0.8, 1.0, |
| | KNN 7 class | 0.8, 0.0, 0.6, 0.1, 0.7, 0.4 | 0.8, 1.0, 0.8, 1.0, 1.0, 0.9 |

Parametric spectral feature set Sensitivity and Specificity table

| Channel | | Sensitivity | Specificity |
|---|---|---|---|
| Oz-Cz | RF 3 class | 0.9, 0.9, 0.5 | 0.9, 0.8, 1.0 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.9, 0.9, 0.6 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.6, 0.7 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Oz-Fpz | RF 3 class | 0.9, 0.8, 0.5 | 0.9, 0.8, 1.0 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.9, 0.8, 0.5 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.6, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Oz-C3 | RF 3 class | 0.9, 0.9, 0.5 | 0.9, 0.8, 1.0 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.9, 0.9, 0.6 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.6, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Oz-C4 | RF 3 class | 0.9, 0.9, 0.5 | 0.9, 0.8, 1.0 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.9, 0.9, 0.6 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.9, 0.1, 0.8, 0.1, 0.6, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Cz-Fpz | RF 3 class | 0.9, 0.9, 0.5 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.9, 0.6 | 0.9, 0.9, 0.9 |
| | KNN 7 class | 0.9, 0.1, 0.8, 0.1, 0.6, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Cz-C3 | RF 3 class | 0.8, 0.8, 0.5 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.9, 0.5 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.8, 0.1, 0.8, 0.1, 0.7, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Cz-C4 | RF 3 class | 0.9, 0.8, 0.5 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.9, 0.6 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.9, 0.1, 0.8, 0.1, 0.7, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Fpz-C3 | RF 3 class | 0.8, 0.8, 0.5 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.9, 0.6 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.8, 0.1, 0.8, 0.1, 0.7, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Fpz-C4 | RF 3 class | 0.9, 0.8, 0.5 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.6, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.9, 0.6 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.9, 0.1, 0.8, 0.1, 0.7, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| C3-C4 | RF 3 class | 0.8, 0.9, 0.5 | 0.9, 0.8, 0.9 |
| | RF 7 class | 0.9, 0.1, 0.7, 0.2, 0.7, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.8, 0.9, 0.6 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.8, 0.1, 0.8, 0.2, 0.8, 0.6 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |

CWT spectral feature set Sensitivity and Specificity table

| Channel | | Sensitivity | Specificity |
|---|---|---|---|
| Oz-Cz | RF 3 class | 1.0, 0.9, 0.6 | 1.0, 0.9, 1.0 |
| | RF 7 class | 1.0, 0.2, 0.8, 0.3, 0.7, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 1.0 |
| | KNN 3 class | 0.9, 0.9, 0.4 | 0.9, 0.8, 1.0 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.7, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Oz-Fpz | RF 3 class | 1.0, 0.9, 0.6 | 1.0, 0.9, 1.0 |
| | RF 7 class | 1.0, 0.2, 0.8, 0.3, 0.7, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 1.0 |
| | KNN 3 class | 0.9, 0.9, 0.3 | 0.9, 0.7, 1.0 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.7, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Oz-C3 | RF 3 class | 0.9, 0.9, 0.6 | 1.0, 0.9, 1.0 |
| | RF 7 class | 1.0, 0.2, 0.8, 0.3, 0.8, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 1.0 |
| | KNN 3 class | 0.9, 0.9, 0.4 | 0.9, 0.8, 1.0 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.2, 0.7, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Oz-C4 | RF 3 class | 1.0, 0.9, 0.6 | 1.0, 0.9, 1.0 |
| | RF 7 class | 1.0, 0.1, 0.8, 0.3, 0.8, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 1.0 |
| | KNN 3 class | 0.9, 0.9, 0.4 | 0.9, 0.8, 1.0 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.2, 0.7, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Cz-Fpz | RF 3 class | 0.9, 0.9, 0.7 | 1.0, 0.9, 1.0 |
| | RF 7 class | 0.9, 0.2, 0.8, 0.3, 0.7, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 0.9 |
| | KNN 3 class | 0.7, 0.8, 0.3 | 0.9, 0.7, 1.0 |
| | KNN 7 class | 0.8, 0.1, 0.7, 0.1, 0.6, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Cz-C3 | RF 3 class | 0.9, 0.9, 0.6 | 0.9, 0.9, 1.0 |
| | RF 7 class | 1.0, 0.1, 0.8, 0.2, 0.7, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 1.0 |
| | KNN 3 class | 0.8, 0.9, 0.3 | 0.9, 0.7, 1.0 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.7, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 1.0 |
| Cz-C4 | RF 3 class | 0.9, 0.9, 0.6 | 0.9, 0.9, 1.0 |
| | RF 7 class | 1.0, 0.1, 0.8, 0.2, 0.7, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 1.0 |
| | KNN 3 class | 0.8, 0.9, 0.3 | 0.9, 0.7, 1.0 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.7, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 1.0 |
| Fpz-C3 | RF 3 class | 0.9, 0.9, 0.7 | 1.0, 0.9, 1.0 |
| | RF 7 class | 1.0, 0.2, 0.8, 0.3, 0.8, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 1.0 |
| | KNN 3 class | 0.8, 0.9, 0.3 | 0.9, 0.7, 1.0 |
| | KNN 7 class | 0.8, 0.1, 0.7, 0.1, 0.6, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| Fpz-C4 | RF 3 class | 0.9, 0.9, 0.7 | 0.9, 0.9, 1.0 |
| | RF 7 class | 1.0, 0.2, 0.8, 0.3, 0.7, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 1.0 |
| | KNN 3 class | 0.8, 0.9, 0.3 | 0.9, 0.7, 1.0 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.1, 0.6, 0.4 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |
| C3-C4 | RF 3 class | 0.9, 0.9, 0.6 | 0.9, 0.9, 1.0 |
| | RF 7 class | 0.9, 0.1, 0.8, 0.3, 0.8, 0.7 | 0.9, 1.0, 0.9, 1.0, 1.0, 1.0 |
| | KNN 3 class | 0.8, 0.9, 0.4 | 0.9, 0.8, 0.9 |
| | KNN 7 class | 0.9, 0.1, 0.7, 0.2, 0.8, 0.5 | 0.9, 1.0, 0.8, 1.0, 1.0, 0.9 |