IDIAP COMMUNICATION REPORT

GENERALIZABLE AUTOMATIC CLASSIFICATION OF SLEEP STAGES

Samuel Michel

Idiap-Com-02-2023

AUGUST 2023

Centre du Parc, Centre du Parc, Rue Marconi 19, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Generalizable Automatic Classification of Sleep Stages

Master Thesis
presented on June 18, 2023
to the Biosignal Processing Research Group
Idiap Research Institute
by

Samuel MICHEL

under the supervision of:

Dr. André Anjos, project supervisor
Flavio Tarsetti, company supervisor

Student number: 21-697-388
Martigny, IDIAP, 2023

Acknowledgements

I am deeply grateful to Dr. André Anjos for his invaluable guidance, expertise, and unwavering support as my master thesis supervisor. His mentorship has been instrumental in shaping the direction of this research.

I would also like to extend my sincere thanks to Flavio Tarsetti, my company supervisor, for his continuous technical support and valuable insights throughout the course of this thesis.

Additionally, I am grateful to Doctor Olivier Pallenca, a respected expert in our field, for his valuable contributions and the wealth of knowledge he shared.

Lastly, I would like to express my appreciation to Prof. Manuel Günther and Gabriele Brunini for their collaboration during this project.

S.M.

Abstract

The gold standard to diagnose sleep disorders is called polysomnography (PSG). A PSG consists in sleeping one or several nights, at a hospital or a sleep center, while wearing different sensors continuously measuring various temporal data (e.g. electroencephalograms, electrocardiograms, electromiograms, oxymetry, respiration rate, etc.). These data are then used by an expert to annotate the PSG (hypnograph) into the differente sleep phases (paradoxal summation, light, moderate and deep sleep). The hypnograph is then used for sleep disorder diagnosis.

The manual annotation process is affected by human limitations: it is time consuming, tedious, not reliable, sensitive to the setup of the different clinics, and to motion noise. Indeed, each sleep center defined his own setup for the PSG. Moreover, it happens that one data is lost due to a motion of the patient during the night (noisy data). Regarding the reliability different studies have shown that for the same PSG two experts may annotated differently.

The aim of this work is to investigate the possibility to automate the classification of PSG into the different sleep phases using machine learning. The main concern will focus on the capacity of such algorithms to be faster, and more reliable than manual scoring. To perform this study, two follow-up questions will gravitate around the main scientific question. We will focus on models which are robust to the setup of different clinics, noise and are fair to different populations. One of the steps of our work is therefore to analyse the ability of an automated classifier to manage data coming from different sleep centers.

We scoped this study to stateless models that do not take into account temporal context. We investigated both hand-crafted and learnable feature extractors. In terms of intra-database performance, our best model was the CNN Chambon model proposed by Chambon et al. in their paper [1]. However, when evaluating generalization across different setups, the random forest model with manually chosen features described in the same paper [1] emerged as the best model.

Table of contents

List of Figures

List of Tables

List of Abbreviations

AASM	The American Association of Sleep Medicine
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
EEG	ElectroEncephaloGram
EMG	ElectroMyoGram
EOG	ElectroOculoGram
FIR	Finite Impulse Response
FN	False Negative
FP	False Positive
LER	Linked Ears Reference
LSTM	Long-Short-Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron
NREM	Non-Rapid Eye Movement
PSD	Power Spectral Density
PSG	PolySomnoGraphy
R&K	Rechtschaffen and Kales
REM	Rapid Eye Movement
RNN	Recurrent Neural Network
TN	True Negative
TP	True Positive

1 Introduction

Sleep remains a domain with ongoing research and many unanswered questions. The study of sleep is a pretty young science as the first observation of brain activity during sleep was realized in 1937 Loomis, Harvey, and Hobart [2]. Since then, extensive scientific research has been conducted in this field, significantly advancing our understanding of sleep. These studies also helped to improve comprehension in other fields like neurology, psychiatry, etc. However, a lot of hypotheses are still under investigation and research continues as aspects of sleep remain mysterious especially sleep disorders.

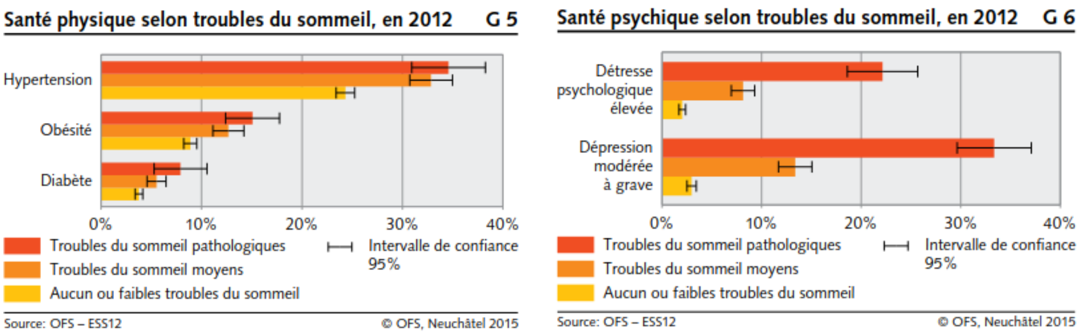
We are aware and sometimes experiment, that a lack of sleep affects many aspects of our life as one of the sleep functions is the regulation of different systems in our body (temperature, metabolism, memorization, etc). The amount of sleep we get every night has short-term consequences for example on our humor, productivity, and energy, but it can also have long-term impacts, especially on health.

According to an article by the Swiss Federal Statistical Office, a quarter of the Swiss population suffers from sleeping disorders [3]. The article also explains the physical and mental consequences of sleep disorders. Due to the number of sleep's biological functions, a lack of sleep is a risk factor for the development of physical diseases like diabetes and obesity (Figure 1.1a). Mental health is not spared as sleep also plays a role in well-being and mental balance. Sleep troubles can lead to mental unbalance such as depression or high mental distress (Figure 1.1b).

The causes of sleep disorders can come from different sources. The most common are [4]:

- **Sleep apnea** is caused by interruptions in breathing during sleep, leading to awakenings. The two main causes of sleep apnea are airway obstruction or a failure of the brain to initiate the breathing process.

- **Restless legs syndrome** is characterized by uncomfortable leg movements or shaking during sleep, which can cause awakenings or difficulties in falling asleep.



(a) Physical health regarding sleep problems

(b) Mental health regarding sleep problems

Figure 1.1: Figure 1.1a illustrates the relationship between sleep disorders and the prevalence of physical diseases such as hypertension, obesity, and diabetes. It demonstrates that as the level of sleep disruption increases, the percentage of individuals affected by these diseases also tends to increase. Similarly, Figure 1.1b presents data on mental health, specifically high mental distress and depression. It reveals that the risk of experiencing mental imbalances is higher among individuals with sleep disorders. The x-axis represents the percentage of individuals suffering from the respective disease within the population experiencing sleep problem. (Source [3]).

- **Insomnia** is defined as the difficulty to stay or/and fall asleep. The causes of insomnia are often linked to environmental conditions (stress, jet lag,...), lifestyle (alcohol, nicotine,...), or mental issues (depression, trauma,...).
- **Sleep behavior disorder** has the effect of physically or vocally acting dreams while asleep.
- People suffering of **Narcolepsy** experience excessive sleepiness during daytime. They may fall asleep even while working.

Sleeping is an oscillation between different levels of depth [5]. Figure 1.2 shows an example of the architecture of one-night sleep in a healthy individual. Typically, the night of a healthy person is composed of three to six cycles that are a sequence of different levels of depth. People suffering from sleep disruption have a different sleep architecture. By analyzing sleep cycles, medical teams establish a diagnosis and possible treatments. The change of state of falling asleep is behavioral and, therefore, can be recorded by simple observation. However, to measure the depth of sleep, and thus the cycles, a medical exam is required.

The most common medical exam to diagnose the cause of sleep disorders is called polysomnography [7] (PSG). A PSG records sleep depth and requires sleeping one or several nights, at a hospital or a specialized clinic. During these nights, different sensors are disposed on the patient's body to measure various data. The most frequently used systems are an electroencephalogram (EEG), which records the electrical activity of the brain, an electrooculogram (EOG), to measure the electrical activity of the eyes, and an electromyogram (EMG) to detect

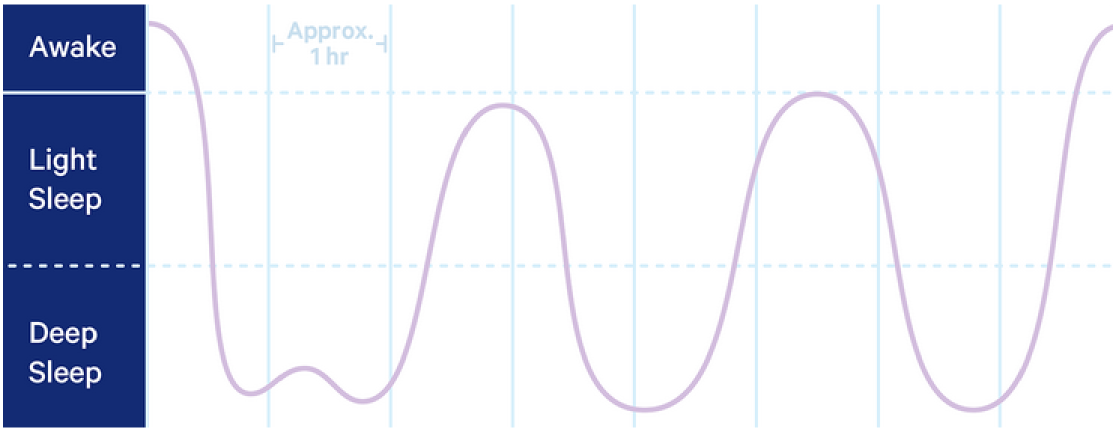


Figure 1.2: One-night sleep of a healthy person. Cycles of the different depth levels of sleep: REM denotes paradoxical summation and N1, N2, N3 are respectively light, moderate, and deep sleep. (Source [6])

muscle tone. Pulse-oximetry, airflow, and other health parameters may also be measured. Figure 1.3 shows an example of a PSG setup and a sample of the curves for each measured data that can be observed by the medical team. An important point to note is that the acquisition of data is a process that often results in the production of noisy or imprecise data. For example, a sensor may detach during the recording period, due to the sudden movement of the patient.

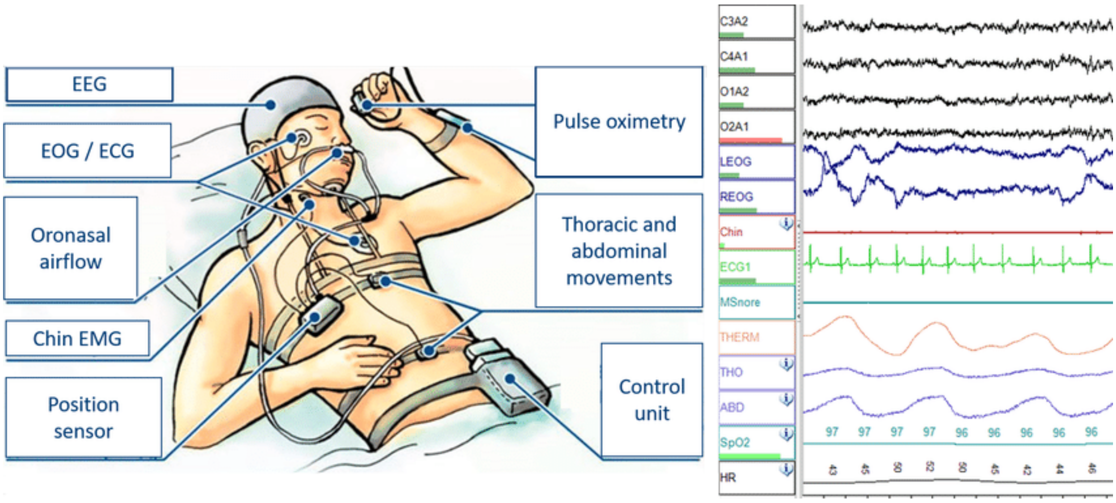
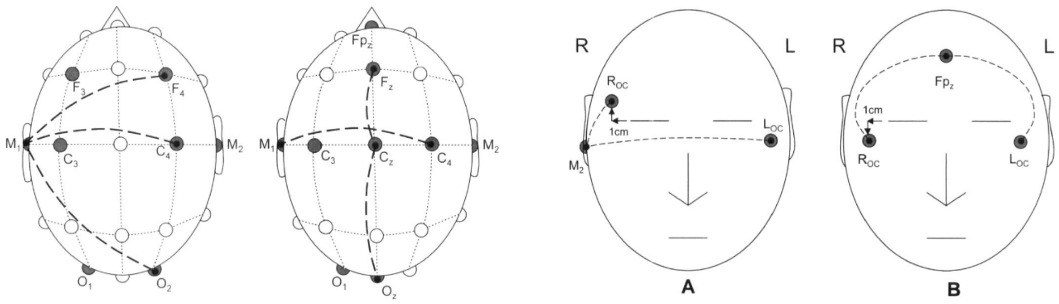


Figure 1.3: Example setup for a PSG medical exam, and a sample of the resulting curves for the different data which are recorded. (Source [8])

Clinics performing PSG recording have their own preferences for recording patient data, especially regarding the types of measured data, and how to position sensors. Figures 1.4a and 1.4b show, for example, two possible positions for the EEG and EOG sensors.

The resulting PSG graph is then used by experts to build the the so-called *hypnogram*, of sleep cycles, displaying different levels of sleep depth. To achieve this, experts are trained to



(a) Two possible setups for EEG sensors

(b) Two possible setups for EOG sensors

Figure 1.4: Different positions for the various sensors are possible. (Source [9])

recognize characteristic patterns on the PSG, and then classify continuous data into discrete phases. We will refer to this as the "annotation of sleep phases". An example of this work is presented in Figure 1.5. Specific patterns of the PSG data (left side of the figure) are classified into 5 phases: awake, paradoxical summation (REM), light (N1), moderate (N2), and deep sleep (N3). The resulting hypnogram is then used by the medical team to diagnose sleep disorders.

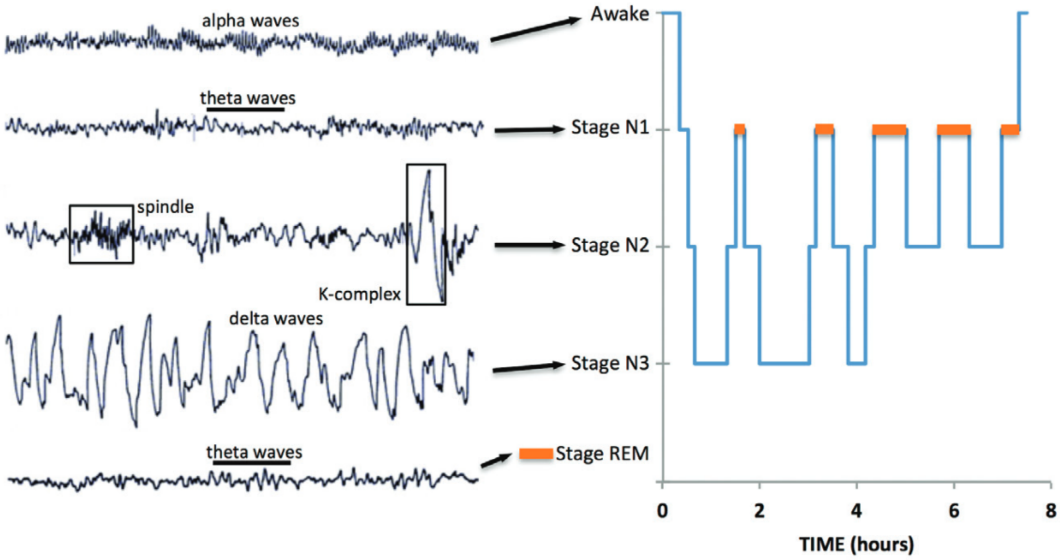


Figure 1.5: Hypnogram (right side) of one-night sleep of a healthy person. The hypnogram is the interpretation of the PSG data (left side) by an expert, *annotating* the different sleep phases. REM denotes paradoxical summation, and stages N1 to N3 respectively describe light, moderate, and deep sleep. The characteristic waves (from the EEG) of every phase are represented on the left. (Source [5])

Two main standard rules are used by healthcare professionals to perform the annotation of sleep phases [5, 10]. The American Association of Sleep Medicine (AASM) method [11], and the Rechtschaffen and Kales method [12].

1. The American Association of Sleep Medicine (AASM) [11] proposes to split the PSG data in windows of 30 seconds. For each sequence, the professional assigns one of the five following states: awake, REM (scientific term of paradoxical summation), and stages N1 to N3 (for light, moderate and deep sleep).
2. On the other hand, the process of the Rechtschaffen and Kales (R&K) [12] method is to split the data either in windows of 20 or 30 seconds. For this technique, deep sleep is subdivided into 2 phases: stages 3 and 4. Sleep cycles are then built using 6 phases: awake, REM, and stages 1 to 4.

The first step of annotation is to distinguish the presence of rapid eye movement (REM) or not (NREM), on the PSG. The REM phase is characterized by fast movement of the eyes under closed eyelids, and a brain activity similar to that present when one is awake. On the other hand, stages N1 to N3 (AASM) or stages 1 to 4 (R&K) are all part of a grand category named "Not Rapid Eye Movement" (NREM), as fast eye movement stopped and brain activity has become lower, as well as muscle tone. On the left side of Figure 1.5, the specific waves (electrical brain activity from EEG) of every phase are represented. The waves are differentiated by their amplitude and frequency. Technical specifications of each type of wave are detailed in section 2.1. As an overview, Figure 1.5 shows that the N1 stage and REM stages contain theta waves, but are distinguished by the presence or not of REM. Stage N2 is characterized by the presence of abnormal wave shapes called spindles and K-complex. Deep sleep is marked by the presence of delta waves (high amplitude and low frequency).

The manual annotation process is affected by human limitations: it is time-consuming, tedious, not reliable, sensitive to the setup of different clinics, and to motion noise. Indeed, we already exposed the problem link with the setup and noisy data. Regarding reliability, different studies have shown that for the same PSG two experts may annotate differently. The authors of [13] compared the annotations of two experts on 196 PSG. Their results show a level of agreement of about 77% which is not very high, especially for a medical exam. This difficulty of a high degree of agreement is in part due to similarity in certain waveforms.

In recent times, the advancement of automated systems for sleep phase annotation relies heavily on artificial intelligence (AI) algorithms. Specifically, machine learning (ML) plays a crucial role in developing computer-based algorithms using mathematical and statistical models. These algorithms are favored for their adaptability and ability to make decisions independently, without human intervention. The primary objective of employing such systems is to automate repetitive tasks and assist humans in the decision-making process.

The temporal aspect of sleep has prompted the development of automated systems that utilize tools such as artificial neural networks (ANN) specifically designed for handling sequential data. Given that polysomnography (PSG) data is temporal and sequential in nature, recent automatic sleep staging systems have incorporated these algorithms. Chapter 2 provides a comprehensive overview of the current state of the field. When citing relevant works, the most commonly employed ANN architectures in this domain are convolutional neural networks (CNN) and recurrent neural networks (RNN) [14].

The aim of this work is to investigate the possibility to automate the classification of PSG into the different sleep phases using ANN algorithms. The main concerns will focus on the capacity of such algorithms to be faster, and more reliable than manual scoring. To perform this study, two follow-up questions will gravitate around the main problem.

We will study algorithms that are robust to the setup of different clinics. Indeed, there is no standard for the setup of PSGs, especially concerning the number of used sensors, and their position. One of the cornerstones of our work will be to analyze the ability of an automated classifier to manage data coming from different sleep centers. This, in turn, implies we are looking for solutions that are robust to multiple factors: types of sensors deployed, quantity, quality, and, finally, positioning during acquisition. Furthermore, we will search for algorithms that are naturally robust to noisy and missing data, related to the nature of the exam.

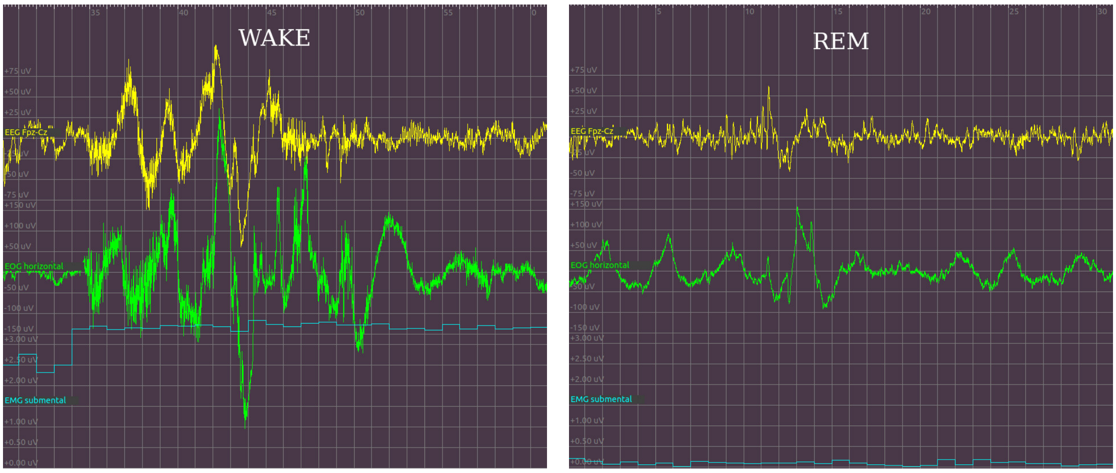
2 State of the art

2.1 Annotation techniques and waveform

Manual annotation of sleep phases is based on criteria regarding wave shapes of PSG signals (e.g. EEG, EOG, EMG), and is typically conducted in 30-seconds windows. The complete process of annotation can be fairly technical. In this review, we focus on the most frequent types of waves, namely the ones which are typical of different sleep stages. We refer to Figure 1.5 for the location and depth of the various sleep phases:

- **Alpha waves** are observed on EEG and are characterized by a frequency between 8-13 Hz. Alpha waves are present when one is awake.
- **Theta waves** are observed on EEG, and begin during stage 1, they have a frequency between 4-7 Hz.
- **Delta waves** are observed on EEG waves and are characteristic of stages 3 and 4. They are also called "slow waves" as they have a frequency between 0.5-3 Hz, and a large amplitude ($\geq 75\ \mu V$).
- **Spindles** are a sequence of abnormal waveforms from 12 to 14 Hz lasting more than 0.5s. They usually occur during stage 2.
- **K-complex** are sharp negative waves followed by smooth positive waves, to be classified as k-complex the duration of this event has to be longer than 0.5s.

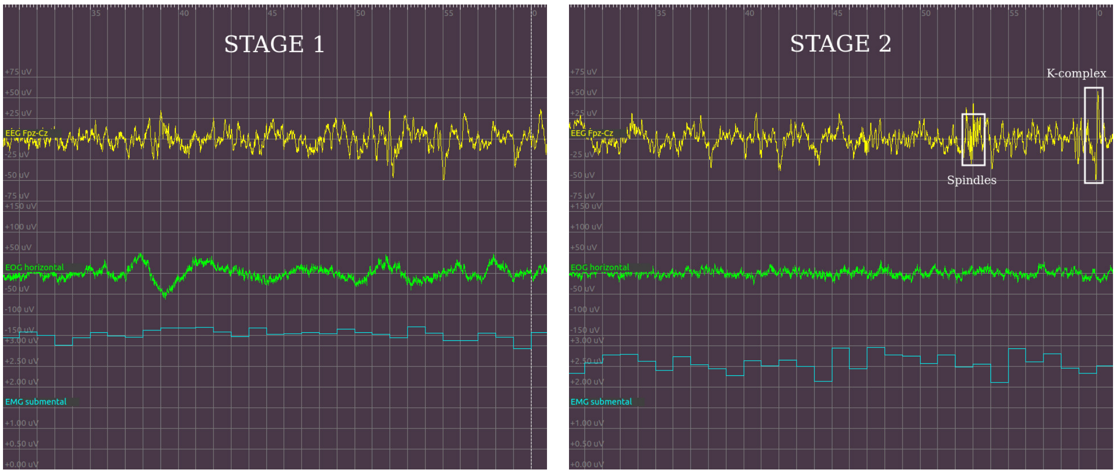
To illustrate the wave shapes on polysomnography (PSG) recording, we present 30-second windows (referred to as "epochs") of each sleep phase in Figures 2.1, 2.2, and 2.3. In each figure, the following signals are represented: 1 EEG (yellow), 1 EOG (green), and 1 EMG (blue). It is worth noting that not all the wave shapes described earlier are present in every sleep phase, as observed in these figures. The typical characterization of each stage using the R&K method is described below:



(a) 30s-window of a PSG labeled as "wake".

(b) 30s-window of a PSG labeled as "REM".

Figure 2.1: The "Wake" label in Figure 2.1a is given due to the high variation on EEG (yellow) and EOG (green). Tonus muscle from electromyography (EMG, light blue, bottom), also presents high values indicating typical awake tension. On the right, at Figure 2.1b, one can, in contrast, observe slower waves on EEG (yellow), REM on EOG (green), and EMG (light blue) is at its lowest value, indicating a relaxed state. These signals come from the EDF file SC4001E0 of the Sleep-EDF dataset and are displayed using EDFBrowser software.



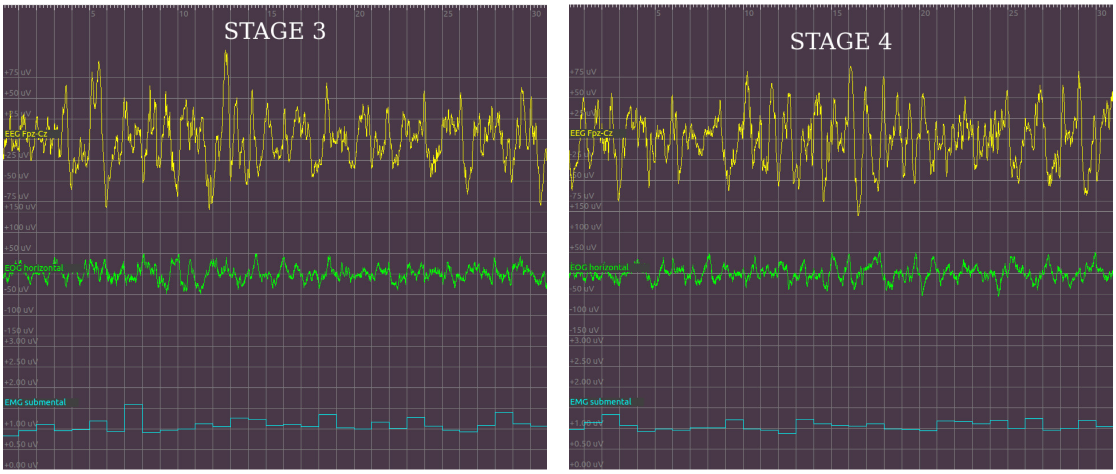
(a) 30s-window of a PSG labeled as "Stage 1".

(b) 30s-window of a PSG labeled as "Stage 2"

Figure 2.2: Figure 2.2a shows a mix of Alpha and Theta waves on EEG (yellow). The variation in EOG is less frequent. Muscle tonus (observed from the EMG signal, light blue) starts to decrease in comparison to wake phase values. On the right-hand side, Stage 2 (Figure 2.2b) is characterized by the presence of Spindles and K-complex waves on EEG, a rather stable EOG signal, and less muscle tonus observed on EMG. These signals come from the EDF file SC4001E0 of the Sleep-EDF dataset and are displayed using EDFBrowser software.

- The **Wake** (Figure 2.1a) phase is characterized by the presence of Alpha waves on the EEG, REM on the EOG, and (muscle) activity on the EMG.
- **REM** (Figure 2.1b) is associated with theta or alpha waves on the EEG, REM on the EOG and the value of the EMG signal is at its lowest level.
- During **Stage 1** (Figure 2.2a) the amount of Alpha waves decreases as Theta waves begin to take place on the EEG. The amplitude of EOG, and muscle movement measured via the EMG begin to decrease as well.
- **Stage 2** (Figure 2.2b) is distinguished by the presence of Spindles and K-complex on the EEG. Alpha waves are not present anymore. The variation in the EOG and EMG are rare.
- **Stage 3** (Figure 2.3a) is marked by a mix of Theta and Delta waves on the EEG. The Delta waves are present between 20 and 50% of the time. There is no variation in the EOG, and muscle activity (measured via EMG) is low.
- **Stage 4** (Figure 2.3b) is characterized by a mix of Theta and Delta waves on the EEG. Delta waves are present more than 50% of the time. There is no variation in the EOG, and muscle activity (EMG) is low.

Description of AASM sleep states can be inferred from this description. The most significant change in AASM labeling rules concerns the combination of Stages 3 and 4 in one single state, called "N3". All other aspects are similar.



(a) 30s-window of a PSG labeled as "Stage 3"

(b) 30s-window of a PSG labeled as "Stage 4"

Figure 2.3: We observe at Stage 3's graph (Figure 2.3a), Delta waves appearing on the EEG (less than 50% of the time), the absence of information on EOG. Muscle tone (captured from EOG) is at a low value. For "Stage 4" (Figure 2.3b), the characterization is similar, however, Delta waves are present more than 50 % of the time on the EEG signal. These signals come from the EDF file SC4001E0 of the Sleep-EDF dataset and are displayed using EDFBrowser software.

2.2 Manual annotation limitations

The two primary metrics used to assess the level of agreement between two scorers for sleep stage classification are Cohen’s Kappa and the intra-class correlation coefficient (ICC) (See Raadt et al. [15] and Koo and Li [16] for more information). Additional indicators, such as F1 and Fleiss Kappa, also exist, but they are less commonly found in the literature.

Cohen’s kappa indicator is relevant to compare categorical variables, which in this case can be used to compare the annotation of both scorers for every 30s-window. The evaluation of this measure gives a result between 0 and 1, that can be compared to a scale (Table 2.1) on the degree of agreement.

Cohen’s kappa		ICC	
Cohen’s kappa value	level of agreement	ICC value	level of agreement
0-0.20	slight	0-0.5	poor
0.21-0.40	fair	0.51-0.75	moderate
0.61-0.80	moderate	0.76-0.90	good
0.81-1	almost perfect	0.91-1	excellent

Table 2.1: Interpretation of the level of agreement between two scorers for Cohen’s kappa and ICC values.

On the other hand, the ICC is used to measure the degree of agreement for continuous quantitative variables. To transform epoch data to continuous variables, the time of every phase is considered. For example, for both technicians, the number of epochs annotated as stage 1 is summed up, and multiplied by the amount of time passed until that epoch. Then, both results are compared. Like with Cohen’s Kappa, the result of the evaluation of this measure lies in the range $[0,1]$. The scale of agreement, however, is defined differently (see Table 2.1).

Several studies have shown the difficulty for different annotators to get a high level of agreement while classifying sleep stages on the same PSG [13, 17–32]. The majority of these studies analyze the inter-rater reliability, some others measure the intra-rater reliability. Inter-rater reliability studies evaluate the level of agreement between two experts scoring the same PSG. The intra-rater studies analyze the ability of single annotators to repeat their scoring.

Whitney et al. [24] analyzed the intra-rater reliability and measured an overall κ, ranging from 0.81 to 0.87. Their study involved three technicians scoring 20 PSGs using the R&K method. Although this study had a small sample size of PSGs and technicians, it stands out as one of the rare studies in the literature where scorers had to label the same PSG twice. It is interesting to note the intra-rater variability and its dependence on the stage being annotated.

Danker-Hopfe et al. [13] measured an overall inter-rater Cohen's Kappa of $\kappa = 0.6816$ between 8 sleep centers. This study is important because of the sheer number of technicians involved, and the number of PSGs annotated. Authors compared the score of ninety-six PSGs manually scored by sixteen experts with the R&K method. The overall inter-score was computed considering five stages: NREM1, NREM2, SWS, REM, and Wake. It fell to $\kappa = 0.6534$ when NREM3 and NREM4 were treated separately.

In 2007, a literature review was performed on the inter-rater reliability which lead to the proposition of current AASM annotation rules with the aim to increase the level of agreement between experts [31]. However, the authors of the review concluded that: "No visual-based scoring system will ever be perfect, as all methods are limited by the physiology of the human eye and visual cortex, individual differences in scoring experience, and the ability to detect events viewed using a 30-second epoch."[1]

A more recent work by Younes et al. [17] showed that the use of AASM method may increase the global scoring reliability as also showed in the paper of Danker-Hopfe et al. [29] (overall κ = 0.76). However, for some stages, the agreement remains low. They measured the inter-rater reliability of 10 technicians on 70 PSGs annotated with the AASM method. On average they got the following ICC for each state: awake = 0.84 (0.70-0.96), N1 = 0.69 (0.30-0.86), N2 = 0.65 (0.37-0.86), N3 = 0.63 (0.18-0.90), and REM = 0.75 (0.58-0.89).

Most studies arrived at the conclusion that the manual annotation of the sleep stages is not reliable. The AASM committee went a step further by claiming that no visual scoring techniques will ever get a high agreement.

2.3 Comparison of manual and automated scoring

With the arrival of computer-based scoring techniques (fully automated or computer-assisted), research was performed to compare the inter-rater reliability of manual vs. computer-based scoring. A review by Penzel et al. [33] from 2013, analyzed the result of 119 publications on the subject and concluded that automatic sleep staging system is still at the beginning of its development and can, therefore, be improved.

2.4 Computer-Assisted Scoring

First algorithms developed to perform automatic sleep stage scoring used more classic ML techniques [14]. All Machine Learning (ML) algorithms follow a similar workflow which is based on the extraction of information (features) in the data. The pipeline of these kinds of

[1] Silber et al. [31], p. 129

algorithms can be described by the following steps:

- **Data preprocessing**: prepares the data to be classified by rescaling, transforming it. This phase is also useful to detect noise or abnormal data.
- **Feature extraction**: This step involves extracting relevant information from the raw data. The process can be linear or non-linear. Popular techniques include time domain analysis, frequency domain analysis, and wavelet transform.
- **Feature selection or dimensionality reduction**: reduces the complexity of the problem by choosing the most relevant features and removing the ones which have less impact. A common technique is called principal component analysis (PCA).
- A model is then trained and tested to perform the **classification**.

The emergence of deep model architectures, which have demonstrated superior performance compared to classical machine learning methods, has significantly influenced various domains. Researchers have explored the application of these models in sleep stage classification as well. One such model is the convolutional neural network (CNN). The key distinction from traditional models lies in the fact that the CNN learns the extracted features autonomously, rather than relying on a manual feature engineering process. A review of these algorithms applied to sleep stage classification was conducted by Fiorillo et al. [14], where the majority of the algorithms utilized CNN and RNN (Recurrent Neural Networks) architectures. These models commonly employed various signals derived from PSG. The authors of the article claimed that the future of these models in sleep staging is promising but requires further research and development.

In the next section, we will categorize sleep-scoring algorithms into two groups: stateful and stateless algorithms. Stateless algorithms lack temporal memory, meaning they are unable to consider the preceding or succeeding epochs to derive more accurate labels for the current epoch. On the other hand, stateful algorithms, as the name suggests, possess the capability to infer sleep states by considering the sequence of epochs, thereby preserving past and future memory.

2.4.1 Stateless algorithms

A paper of Boostani, Karimzadeh, and Nami [34] compared the classification of stateless algorithms only based on EEG signals. One of the algorithms in this paper was developed by Fraiwan [35] and is based on the random forest ML model. To realize this work they only used a single EEG channel. For the preprocessing step, they did not use any particular method other than normalizing and filtering the signal (baseline drift). The time-series signal is then split into 30s windows. They decided to use hybrid methods of feature extraction which contains two consecutive analysis:

- **A time-frequency analyzer** which is a tool to compute time-frequency transform of time series. In this paper they implemented three different tools to compare them: Hilbert–Huang Transform, continuous Wavelet transform, and Choi–Williams distribution.

- **Entropy Calculus** which analyses the 2D time-frequency structure to detect the presence of wave shapes described in Section 2.1. They extract seven features because they consider two additional wave types (Beta 1 and 2). The calculation of entropy was done using Renyi's method.

Finally, with the seven features extracted, authors were able to train a Random Forest model using Bootstrapping. They fixed the number of trees at 10 and the number of features per tree at 4. The training and testing were done on 16 PSGs. They got a Kappa coefficient of 0.76 when using the continuous wavelet transform.

Liang et al. [36] proposed an algorithm utilizing a linear classifier machine learning model. The signal was preprocessed using an 8th-order filter. The processed signal was then used in two different ways. Firstly, a multi-scale entropy calculation was performed to extract features from the signal. Secondly, the signal was passed through an 8th-order filter again to extract theta waves. The resulting coefficients were analyzed using an auto-regressive model. The purpose of this second feature extraction, focused on theta waves, was to improve the classification of the N1-stage, which is known to be challenging to identify. A linear classifier was trained and tested using these extracted features. Additionally, eleven manually designed rules were used to refine the final result. The algorithm achieved an overall Kappa coefficient of 0.81 on the Physionet Public Database.

In 2018, Chambon et al. [1] developed a small CNN architecture, as described in their study [1]. The architecture incorporated a CNN layer to perform independent component analysis [37], a linear spatial filtering technique. The model also included a series of CNN and max-pooling layers to extract spectral features from the signals. Subsequently, class probabilities were computed using a dense layer followed by a softmax activation function. The researchers demonstrated that increasing the number of channels and incorporating multi-modal signals such as EEG, EOG, and EMG improved the balanced accuracy of the model. To compare their results, they employed a gradient boosting model that utilized manually chosen extracted features based on an article by Lajnef et al. [38]. Both models were trained and tested on the MASS dataset, specifically focusing on the SS3 subset.

A deeper 1D-CNN network was introduced by Satapathy and Loganathan [39]. The architecture consists of nine convolution blocks, each containing a 1D-convolution layer, a batch normalization layer, a ReLU activation layer, and a max-pooling layer. The model concludes with two dense layers and a softmax activation layer. Similar to previous works, the researchers employed multi-modal channels and achieved an accuracy of approximately 99% on their test set. It is noteworthy that their experiments were conducted using the ISRUC-Sleep dataset.

However, this remarkable accuracy is surprising considering that no other paper, even those exploring deeper models, has achieved such levels of accuracy. The model may be overfitted as they trained and tested the model using only five patients.

2.4.2 Stateful algorithms

Most recent stateful algorithms are recursive algorithms, often based on recurrent neural network architectures (RNNs). These RNN algorithms typically operate directly on raw data, without the need for manual feature extraction, and are capable of processing large amounts of data. One common approach is to combine CNN layers with RNN layers. CNN networks are used to create an embedding representation, while RNNs are able to model temporal context. Recently, researchers have started exploring the application of transformer models, following the work of Vaswani et al. [40], in sleep stage classification. For example, Eldele et al. [41] have begun investigating the use of transformer models in this context.

The "DeepSleepNet" model, proposed by Supratak et al. [42], consists of two CNN networks to extract time-invariant features. The resulting feature vectors from these networks are concatenated and fed into two bidirectional LSTM layers to model the sleep sequence and infer transition rules between epochs. Through 31-fold cross-validation using the MASS dataset, they achieved a Cohen's Kappa of 0.8. The use of two CNN networks aims to capture different types of features, both temporal and spectral.

A similar architecture was designed by Biswal et al. [43], but they incorporated a spectrogram representation as a preprocessing step. The spectrogram of each channel is passed through two 1D-CNN blocks with different sizes. The features from all input channels are then concatenated and fed into a residual network composed of 16 convolutional blocks with 4 skip connections. The output of the residual network is used as input for an RNN. Notably, this work included cross-dataset analysis, where training was performed on the Massachusetts General Hospital Sleep Laboratory dataset and testing was conducted on the Sleep Heart Health Study dataset using 2 EEG channels. In this configuration, they achieved a Cohen's Kappa of 0.73.

Malafeev et al. [10] focused on generalizing from healthy sleep data to patient sleep data. They explored four models: two feature-based models (manually chosen features) using a random forest classifier combined with an HMM and an LSTM, and two models that directly processed raw data using a combination of CNNs for feature extraction and LSTMs to incorporate temporal context. Through intra-database analysis, training, validating, and testing on healthy data, they obtained an overall Cohen's Kappa of 0.8 for all models. However, they observed lower results when training and validating with healthy data and testing on patient data.

2.5 Datasets

To develop automatic algorithms for sleep stage classification there must be annotated PSG data. Different datasets are available from the literature. Most popular databases are listed in Table A.3. There exists a lot of PSG data that are publicly available, which is an interesting point for reproducibility and comparison to previous work. More details on the specific datasets selected for this work are given in the next chapter.

2.6 Scope of This Work

Only few papers actually studied generalization across different setups [10, 43, 44] by performing cross-database analysis. We decided to carry out this work in this direction by designing protocols of evaluation (See Section 3.2.6) based on the generalization across setups, by training and evaluating our models on different datasets.

We decided to scope this work to stateless algorithms by making the hypothesis that stateless architectures are more robust to manual annotation and will be less biased toward different setups and populations.

3 Methods and Data

3.1 Data

3.1.1 EDF Datasets

The EDF dataset consists of two subsets recorded in different years and under different conditions:

- **ST-EDF subset:** This subset was registered in 1994 under medical conditions. It includes two nights' PSG recordings from 22 individuals, one with temazepam intake and the other with a placebo intake.
- **SC-EDF subset:** This subset was obtained between 1987 and 1991. It comprises two 20-hour PSG recordings from 78 individuals. The recordings were performed using a modified walkman while one was pursuing normal activities.

Due to the differences in the methodology used to record the PSGs, we treat both subsets as two distinct datasets.

3.1.2 MASS Datasets

The MASS dataset contains five subsets (SS1 to SS5). For this study, we specifically worked with the third subset, SS3, as it is typically used in the literature [1, 42]. SS3 contains a substantial number of PSGs (62 recordings) and all are healthy patients.

Our choice of datasets was guided by two main criteria: reproducibility and alignment with existing literature. Consequently, we selected the Sleep-EDF and MASS datasets for our study. Both datasets are publicly available, with the caveat that research ethics board approval is required for accessing the MASS dataset. These datasets have been utilized in various previous works, including those by Chambon et al. [1], Supratak et al. [42], Roy et al. [45], Perslev et al. [46], Perslev [47], and Tsinalis, Matthews, and Guo [48].

Table 3.1: Information about the number of subjects, their age, if they are sick or healthy and the method used to annotate the PSG for the different datasets selected for this work.

Datasets	Number of PSG	Age (range)	Subjects	Annotation (method)
ST-EDF (Sleep-EDF)	44	18-79	22 healthy	R&K
SC-EDF (Sleep-EDF)	153	25-101	78 healthy	R&K
SS3-MASS (MASS)	62	20-69	62 healthy	AASM

Table 3.1 offers general information about the datasets used in our study. It includes details such as the number of PSG recordings, the age range of subjects, their health condition, and the annotation method employed to label the PSGs. For this study, we made the choice to work exclusively with healthy patients. This decision ensures that our analysis focuses on a homogeneous group, facilitating clearer insights into the sleep patterns and characteristics of healthy individuals. Additionally, we observe variations in the annotation method across the different datasets.

Additionally, Table 3.2 describes the setup of each dataset used to perform the PSGs. It includes the number of channels for EEG, EOG, EMG, and a list of any additional sensors utilized. By examining this table, we can readily observe the differences in configurations across the various datasets. Another aspect that stands out from the table is the variation in sampling rates used in the recordings. The sampling rate is an essential factor as it directly impacts the temporal resolution of the data.

To assess the generalization of our models across different setups, we aimed to use datasets with diverse configurations. Figure 3.1 illustrates the EEG channel configuration for the selected datasets. While EDF-ST and EDF-SC share the same channels, only three channels are common between MASS (SS3) and EDF (ST and SC). Hence, the chosen datasets fulfill our requirements for testing our hypothesis.

Table 3.2: Information about the setup of the different sensors used while recording the PSGs for each dataset. In parenthesis it is specified the sampling frequency of the channels

Datasets	EEG[h]	EOG	EMG	Other channels
ST-EDF (Sleep-EDF)	3 channels (100)	1 channel (100)	1 channel (100)	EM[b]
SC-EDF (Sleep-EDF)	3 channels (100)	1 channel (100)	1 channel (1)	RS[c],RT[d],EM[b]
SS3 (Mass)	20 channels[e] (256)	2 channels (256)	3 channels (256)	ECG

[a] Fpz-Cz, Pz-Oz (Bipolar channels)

[b] Event marker

[c] Respiratory signal

[d] Rectal temperature

[e] C3, C4, Cz, F3, F4, F7, F8, O1, O2, P3, P4, Pz, T3, T4, T5, T6, Fz, Fp1, Fp2, Oz (referential channels, reference = LER)

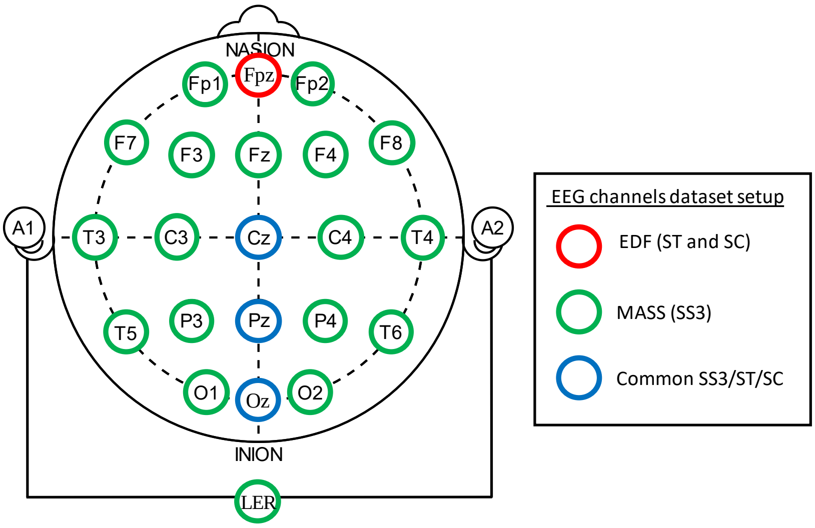


Figure 3.1: EEG channel configuration for the selected datasets.

3.2 Methods

Machine learning problems often follow a common workflow. An overview of this pipeline can be seen in Figure 3.2. The following sections described the different choices and hypotheses we made at each step of this workflow and which constitute our methodology of work. To make this work proceed further and for reproducibility, we developed the different parts described in the previous sections in a Python package named sleepless[1].

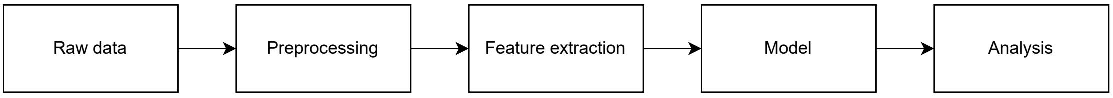


Figure 3.2: Generic machine learning workflow.

Sleepless is a Python package crafted around the principles of reproducibility and machine learning. Its core functionalities are centered on the analysis of sleep data, particularly polysomnography and corresponding labels obtained from various datasets. The package provides encapsulated modules that enable the loading of sleep data, the training and evaluation of machine learning models, and the provision of tools for result analysis.

One of the package's primary strengths lies in its ability to facilitate the training and evaluation of machine learning models using frameworks such as Scikit-learn [49] and PyTorch [50]. These models are specifically tailored to classify different sleep phases, allowing users to gain valuable insights into sleep patterns and associated phenomena.

Beyond the training and evaluation of machine learning models, sleepless offers sophisticated tools for in-depth result analysis. Researchers can explore the outcomes of their models, gaining a deeper understanding of the classification performance and potential areas for improvement.

To maintain the package's reliability and robustness, sleepless incorporates continuous integration and comprehensive testing protocols. These measures ensure that the functionality remains intact across different environments, minimizing the risk of errors or inconsistencies.

To streamline collaboration and version control during development, we leveraged Git and hosted our repository on GitLab. This enabled seamless code management, allowing multiple contributors to work together efficiently.

Additionally, to enhance the usability and accessibility of Sleepless, we provided detailed documentation on how to use the package effectively. This documentation serves as a valuable resource for users, guiding them through the functionalities and implementation details.

Furthermore, to simplify the installation and distribution of the package, we utilized Conda [51] packaging. Conda allowed users to easily install sleepless and manage its dependencies.

[1] https://pypi.org/project/sleepless/

Figure 3.3 illustrates the architecture of Sleepless, comprising encapsulated modules and a command-line application for launching scripts to conduct experiments.

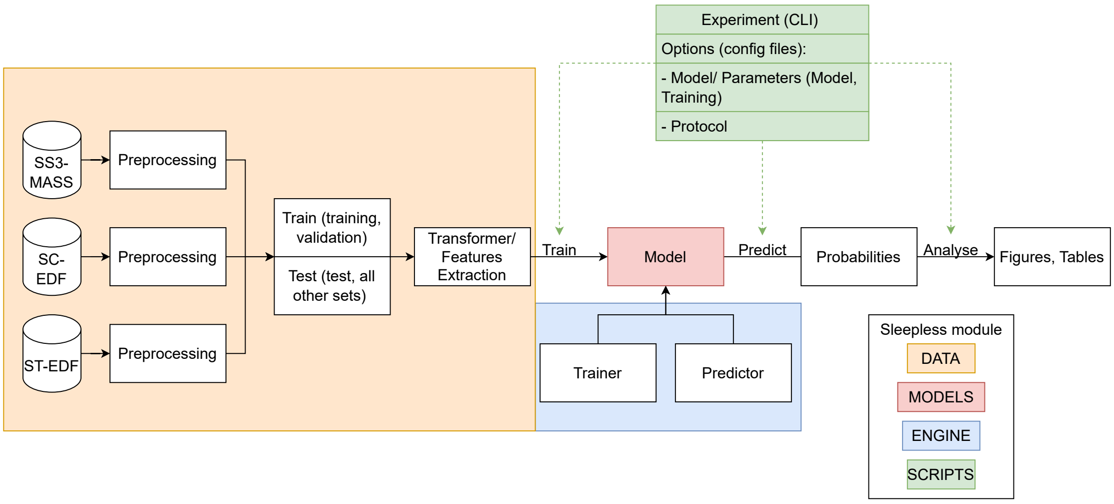


Figure 3.3: Architecture of the sleepless package.

3.2.1 Raw Data Loading

Each PSG comprises three essential components: a raw signal EDF file, an EDF file containing expert annotations of the PSG, and metadata related to the patient, such as age and gender. The loading of these files was facilitated using the MNE-Python package [52, 53], which offers convenient functions for handling EDF files.

To create the three subsets of the datasets (train, validation, and test), we employed a random split approach. The resulting splits were stored as JSON files within the package to ensure reproducibility, making it possible for others to obtain the same subsets for their experiments.

3.2.2 Preprocessing

Label Fusion

Label fusion allows us to effectively manage datasets that might have variations in their annotation rules. For our work, we opted to adopt the AASM method as the standard.

To achieve label fusion, we combined the S3 and S4 phases of the R&K annotation method and mapped them to the N3 phase of the AASM annotation rule, following the mapping presented in Table 3.3. This process assumes a uniform epoch length of 30 seconds across all datasets.

Label Fusion	
AASM Labels	R&K Labels
Wake	Wake
REM	REM
N1	S1
N2	S2
N3	S3 and S4

Table 3.3: Label fusion technique to handle dataset with different annotation method.

Filtering

Following prior works [1, 39, 54, 55], we decided to apply a frequency filter to our data. The frequencies of interest are distributed between 0.3 Hz and 30 Hz, so we chose to use a band-pass filter to selectively retain this specific frequency range.

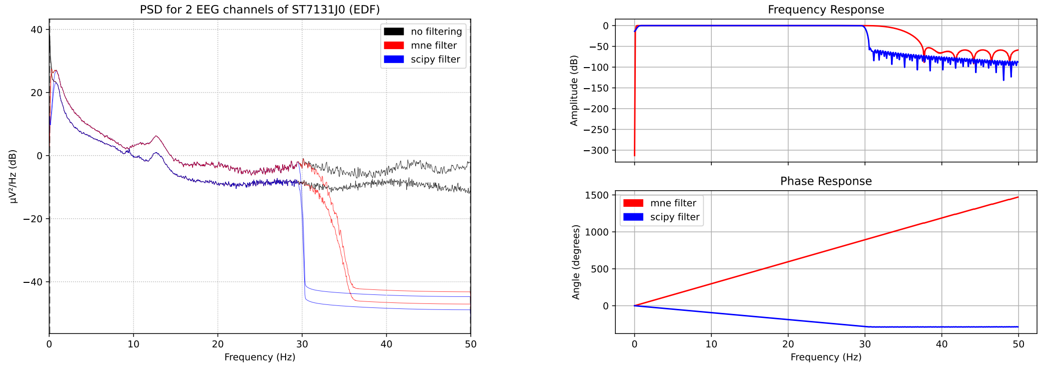
In signal processing applications, two types of digital filters exist: Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). Both types have different characteristics concerning stability, latency, computation cost, and linearity of phases. For our purposes, we opted for FIR filters, as they are widely used in the literature [1, 39, 54, 55]. Notably, we computed the filter forward and backward to create a zero-phase filter, thereby avoiding any signal delays.

Powerline noise can often affect the raw data recorded by the electrodes. A common solution to mitigate this noise is to use a notch filter, an Infinite Impulse Response (IIR) band-stop filter designed to target specific frequency ranges. Powerline noise is typically present at frequencies near 50 Hz (in Europe) or 60 Hz (in North America). However, in our case, we anticipate that a notch filter will not be necessary. This is because we have already applied a band-pass filter to all signals, restricting the frequency range to 0.3 Hz to 30 Hz, effectively excluding the powerline noise frequencies. To confirm this hypothesis, we conducted tests on the data. It is also important to note that the notch filter is not needed for the EDF datasets since the sampling rate was 100 Hz. According to the Nyquist theorem, the highest frequency that can be recorded is half of the sampling rate (50 Hz in this case), and thus, powerline noise at 50 Hz is inherently filtered out.

We conducted a comparison between the implementations of two packages, MNE-Python [52, 53] and Scipy [56], using both EDF and MASS datasets. To perform the comparison, we selected two EEG channels from a file in the EDF-SC and MASS dataset. We then plotted the Power Spectral Density (PSD) of these channels before and after applying the filter in both packages.

Additionally, we visualized the frequency and phase response for each implementation.

For both implementations, we designed a Finite Impulse Response (FIR) pass-band filter with a frequency range between 0.3 Hz and 30 Hz. In Figure 3.4a, we presented the PSD of the two EEG channels before and after applying the filtering process. Simultaneously, Figure 3.4b illustrates the magnitude and phase response of the filters used. It is worth noting that we implemented a zero-phase filter by applying the filter both forward and backward. In both Figures 3.4, we observe that the cutoff frequency of the Scipy filter (blue curves) is closer to 30 Hz compared to the MNE-Python filter. In both cases (red and blue curves), the signal remains unaltered between 0.3 Hz and 30 Hz, as evidenced by the overlap with the signal without any filtering (black curves).



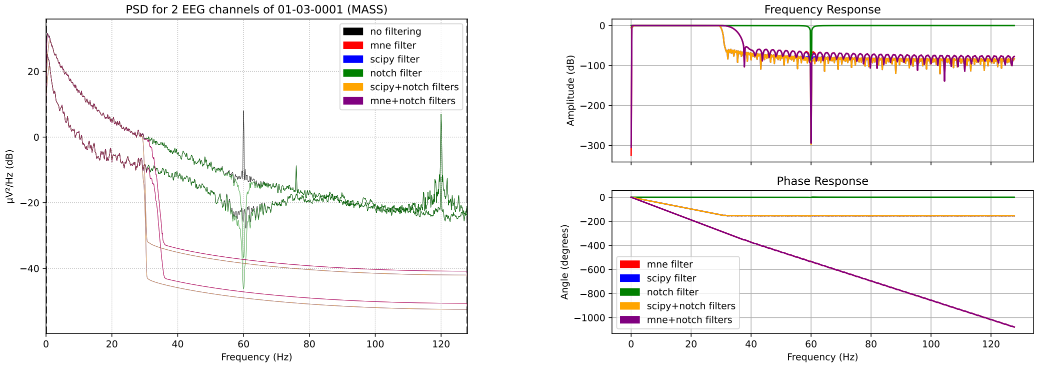
(a) PSD of 2 EEG channels for different filters and without any filtering.

(b) Magnitude and phase responses for different filters.

Figure 3.4: Figure 3.4a illustrates the PSD of 2 EEG channels recorded from the ST7131J0 subject. The black curve represents the PSD without any filtering, while the red curve represents the PSD after applying a zero-phase pass-band filter using the MNE-Python library implementation. Additionally, the blue curve represents the PSD after applying a zero-phase pass-band filter using the Scipy package. In Figure 3.4b, we present the magnitude and phase responses of the filters from both the MNE-Python and Scipy implementations.

We replicated the same process with a PSG from the MASS dataset, and we extended the analysis to include notch filtering to verify the assumption made in Section 3.2.2 (i.e., no notch filter is needed). Figure 3.5a presents the results, which are similar to those observed for the EDF-SC PSG. The signals remain unchanged within the pass-band frequency range, and the cut-off frequency of the Scipy filter is closer to 30 Hz, aligning with our previous observations.

Regarding the notch filter implementation, we observed that it is indeed unnecessary. The PSD of the signal filtered with and without the notch filter is entirely superposed (yellow and blue lines). This further confirms that the band-pass filtering alone successfully eliminates the powerline noise frequencies, rendering the notch filter useless in this case.



(a) PSD of 2 EEG channels for different filters and without any filtering.

(b) Magnitude and phase responses for different filters.

Figure 3.5: The Figure 3.5a displays the PSD of 2 EEG channels recorded from the 01-03-0001 subject. The black curve represents the PSD without any filtering, while the red curve represents the PSD after applying a zero-phase pass-band filter using the MNE-Python library implementation. Similarly, the blue curve represents the PSD after applying a zero-phase pass-band filter using the Scipy package. In addition, the green curve shows the PSD after applying a notch filter to the raw signal. The yellow curve represents a combination of the Scipy pass-band filter and the notch filter, and the purple curve represents a combination of the MNE-Python pass-band filter and the notch filter. In Figure 3.5b, we present the magnitude and phase responses for the different filters.

Channel Interpolation

The objective of channel interpolation is to estimate the electrical potential at a specific location on the scalp where no electrode was present to record it. Although various interpolation techniques are available, we opted to utilize the nearest neighbor interpolation method. While it may not be considered the most advanced interpolation technique according to Perrin et al. [57], it remains popular due to its ease of implementation and continued use in commercial software and toolboxes, as mentioned by Svantesson et al. [58].

We chose this method to ensure the comparison of the same channel across all datasets. Since the MASS dataset lacked recording from the Fpz location (Figure 3.1), we employed nearest neighbor interpolation using the Fp1 and Fp2 locations to estimate the electrical potential at Fpz.

$$V(x, y) = \frac{\sum_{i=1}^{k} v_i d_i^n}{\sum_{i=1}^{k} d_i^n} \tag{3.1}$$

where:

- (x,y) is the location we want to interpolate the electrical potential of

- v_i the electrical potential of the i^{th} neighbour electrode
- d_i the distance between (x,y) and (x_i, y_i) the location of the i^{th} neighbour electrode

The general nearest neighbors interpolation formula for k neighbors and of order n is described by Equation 3.1. As we used only 2 neighbors and given their equidistance, the equation simply becomes the mean of both electrical potentials.

Crop Wake Time

The PSGs of the EDF-SC dataset were recorded for periods exceeding 20 hours. However, the majority of this time captures the participant when they were not sleeping, which is less relevant for our specific goal of classifying sleep stages. To prevent any biases in the model, we applied a cropping process to retain only the sleep periods. We kept a portion of the wake time before and after the sleep period, extending it to 30 minutes on either side.

Bipolar Reference Computation

There are various methods to record the electrical potential for EEG channels. One common approach is to record all electrodes with respect to a common reference, such as the Linked Ears Reference (LER), as depicted in Figure 3.1. Another method involves using bipolar electrodes, which was employed in the EDF datasets. In this case, the electrical potential between two locations on the scalp (e.g., Fpz-Cz, Pz-Oz) is recorded.

The goal of bipolar reference computation is to estimate the electrical potential of brain activity between two recorded electrodes. For instance, if T4 and T3 electrodes were recorded with a LER reference, and we wish to determine the potential between T4 and T3, we can compute it through bipolar reference computation. In this scenario, it simply involves taking the difference between both electrodes, considering the common reference.

Since the EDF datasets were recorded with bipolar electrodes, and the MASS dataset used the LER reference, we applied the bipolar reference computation technique to the MASS dataset. This allows us to obtain the same bipolar electrode configuration as the one present in the EDF datasets, enabling consistent and comparable analysis across the different datasets.

Re-sampling

The different datasets used in our study do not have the same sampling rate. While this variation may not pose a problem when manually extracting features from the data, it becomes an issue when working with a CNN since the filter sizes in the network must be defined with respect to the sampling rate.

To enable the use of different datasets on the same CNN model and ensure consistency in the analysis, we aimed to establish a common sampling rate. Considering that the relevant frequency range of interest lies between 0.3 Hz and 30 Hz, we made the decision to work with a sampling rate of 100 Hz. By resampling the MASS dataset to 100 Hz, we harmonized the sampling rate across all datasets.

3.2.3 Features Extraction

The extracted features used in our work can be selected through manual engineering, where we choose the most relevant ones, or learned automatically using a CNN.

Table 3.4: List of features we worked with; Note: x represents a time-series of N samples, $\bar{x}$ represents the mean of the data, B the number of bin in the PSD and $f_a\text{and}f_b$ the range of the frequency bands

Name	Formula	Type of Features
Mean	$\frac{1}{N}\sum_{i=1}^{N} x(i)$	Temporal
Variance	$\frac{1}{N-1}\sum_{i=1}^{N}(x(i)-\bar{x})^2$	Temporal
Skewness	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x(i)-\bar{x})^3}{\left(\frac{1}{N-1}\sum_{i=1}^{N}(x(i)-\bar{x})^2\right)^{3/2}}$	Temporal
Kurtosis	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x(i)-\bar{x})^4}{\left(\frac{1}{N-1}\sum_{i=1}^{N}(x(i)-\bar{x})^2\right)^{2}}$	Temporal
75% Quantile	75th percentile of the data	Temporal
Total Power[a]	$\frac{\sum_{k=f_a}^{f_b}\text{PSD}[f]}{f_b-f_a}$	Frequency
Relative Power[a]	$\frac{\sum_{k=f_a}^{f_b}\frac{\text{PSD}[f]}{\sum_f \text{PSD}[f]}}{f_b-f_a}$	Frequency
Relative Power Ratio[b]	$\frac{\delta}{\theta},\frac{\delta}{\alpha},\frac{\delta}{\sigma},\frac{\delta}{\beta},\frac{\theta}{\alpha},\frac{\theta}{\sigma},\frac{\theta}{\beta},\frac{\alpha}{\sigma},\frac{\alpha}{\beta},\frac{\sigma}{\beta}$	Frequency
Spectral Entropy[c]	$-\frac{1}{\log_2(B)}\sum_{f=1}^{B}P(f)\log(P(f))$	Entropy

[a] We worked with 5 bands with the following frequency ranges: δ [0.5, 4.5], θ [4.5, 8.5], α [8.5, 11.5], σ [11.5, 15.5], β [15.5, 30]

[b] Ratio of the relative power value of each bands in all 10 possible combinations

[c] $P(f)$ is the relative power value at frequency f

Our feature set consists of three types: time domain features, frequency domain features, and non-linear features (entropy), as detailed in Table 3.4. Utilizing different types of features is beneficial as they offer complementary information. Temporal features provide statistical descriptions of the signal, capturing information about temporal variations. Spectral features enhance periodic patterns, while entropy features measure signal complexity and purity.

In addition to hand-crafted features, we also employed CNNs for learned feature extraction. The principle of a simple CNN architecture is depicted in Figure 3.6. CNNs are highly effective in signal processing, as they utilize convolutional layers to apply a series of filters to the input signal, extracting relevant features through convolution. Each filter is designed to detect specific patterns, and their weights are learned during the training process.

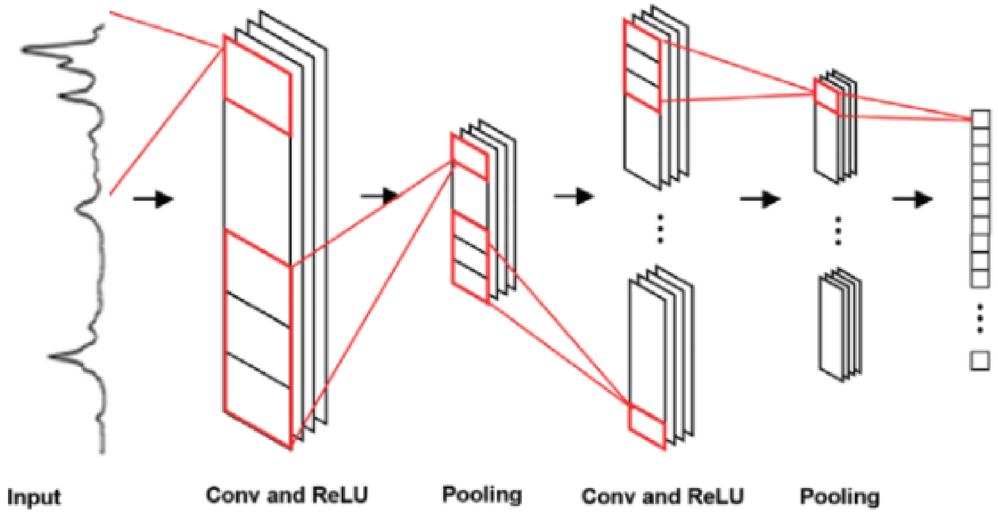


Figure 3.6: Example of CNN architecture, which is a stacking of Convolutional layers with ReLu activation and max-pooling layers. (Source [59])

The convolution layers are typically followed by non-linear activation functions, such as ReLu (Rectified Linear Unit). The presence of these activation functions is crucial to enable the model to capture non-linear features present in the data. By stacking convolution and activation layers, CNNs can simultaneously identify various features at different levels of abstraction, creating feature maps. To reduce the dimensionality of the feature maps and extract the most relevant information, CNNs often incorporate max pooling. This downsampling process reduces computational complexity while preserving essential information.

It is important to note that in our case, the CNN network and the classifier are combined as an end-to-end model to ensure back-propagation through both the CNN and the classification part during model training.

3.2.4 Models

We used two types of models to classify our features, models based on decision-tree theory and multi-layer perceptron (MLP). In the machine learning workflow, the model follows different steps which are directly connected to the way we split our data (train/validation/test):

- **Training** is the process through which an optimization algorithm tunes the internal parameters and weights of the model to minimize the loss function using the training set. The loss function is a mathematical function that measures the performance of the model according to the predictions of the model and actual values.
- **Validation** is used to perform grid-search or other techniques to find the best hyper-parameters, which are parameters not internal to the model but still affect its performance (e.g. learning rate). To choose the right hyper-parameters, the model is evaluated on the validation set.
- During **Evaluation**, all parameters of the model are now fixed, and its performance is reported using the test set.

To perform experiments, we implemented in our package 4 state-of-the-art algorithms, 2 are hand-craft features based and 2 use CNN to extract features.

The hand-craft features based models and their scripts to train and evaluate the model have been implemented with the Scikit-learn library [49]. The feature extraction process was implemented in part with the MNE-Python package [52, 53]. The models using CNN and their scripts to train and evaluate the model have been implemented with the PyTorch library [50].

MNE-Python [1, 52, 53] Tutorial With Random Forest Classifier

Firstly, we developed the algorithms part described in the documentation of the MNE-Python library [1, 52, 53]. This implementation represents a simplified version of the one described in Section 3.2.4. In this approach, we utilized only the relative power of PSD in 5 frequency bands, as outlined in Table 3.4, resulting in 5 features per channel. Following the feature extraction process, we employed a random forest model to classify the samples. A grid search was performed to identify the optimal hyper-parameters for the model. These hyper-parameters are described in 4.1.2.

Chambon CNN [1]

The model described in the paper by Chambon et al. [1] is an end-to-end network composed of a CNN to extract features and a classifier. The model was built to handle both multivariate channels and multi-modal channels.

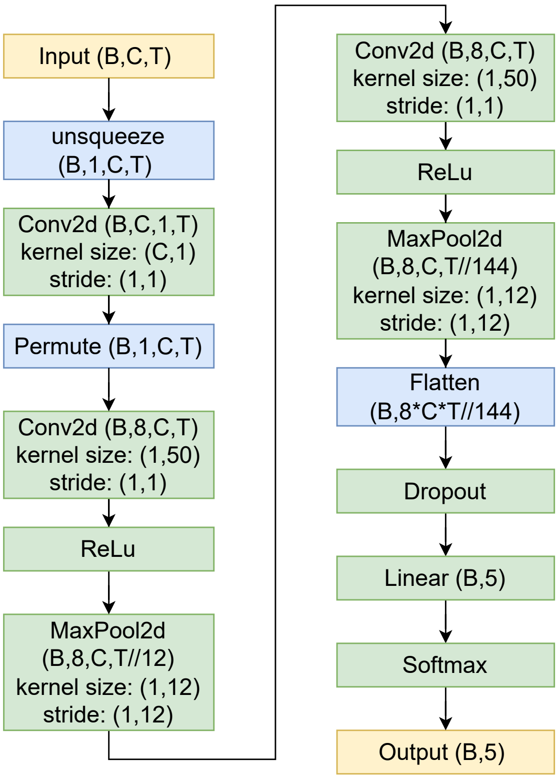

Input (B,C,T)
unsqueeze (B,1,C,T)
Conv2d (B,C,1,T) kernel size: (C,1) stride: (1,1)
Permute (B,1,C,T)
Conv2d (B,8,C,T) kernel size: (1,50) stride: (1,1)
ReLu
MaxPool2d (B,8,C,T//12) kernel size: (1,12) stride: (1,12)
Conv2d (B,8,C,T) kernel size: (1,50) stride: (1,1)
ReLu
MaxPool2d (B,8,C,T//144) kernel size: (1,12) stride: (1,12)
Flatten (B,8*C*T//144)
Dropout
Linear (B,5)
Softmax
Output (B,5)

Figure 3.7: Architecture of the model from Chambon et al. [1] paper. The figure contains the names and the output shape of the different layers. B represents the batch size, C the number of channels, and T the number of data points.

The architecture of the model, as shown in Figure 3.7, consists of three parts:

1. A first 2D-convolutional layer is applied to the channels to perform linear filtering this technique is similar to performing Independent Component Analysis (ICA) [60].

2. Then, two blocks are stacked, each composed of 1 2D-convolutional layer with a ReLU activation function and a max-pooling layer, to extract the spectral features.

3. Finally, a dropout layer is utilized to avoid overfitting.

The classifier is composed of 1 dense layer and one softmax activation function. The dense layer represents a fully connected layer of a Multi-Layer Perceptron (MLP) where the neurons are fully interconnected. The model has a total of 6,033 parameters. For training, we employed the weighted cross-entropy loss and the Adam optimizer algorithm.

Chambon Hand-Craft Features [1] With Gradient Boosting Classifier

In the same paper by Chambon et al. [1], a comparison was made between the CNN model and a hand-crafted feature-based algorithm. This hand-crafted approach employed all the features described in Table 3.4, resulting in a total of 26 features per channel.

For classification, the authors implemented a gradient boosting model using the XGBoost package [61]. A grid search was employed to determine the best hyper-parameters for this model. These hyper-parameters are described in 4.1.2.

Satapathy Nine Layers CNN [39]

Similar to the Chambon CNN, the Satapathy Nine Layers CNN is an end-to-end model comprising a CNN for feature extraction and a classifier. This model is capable of handling both multi-variate and multi-modal channels.

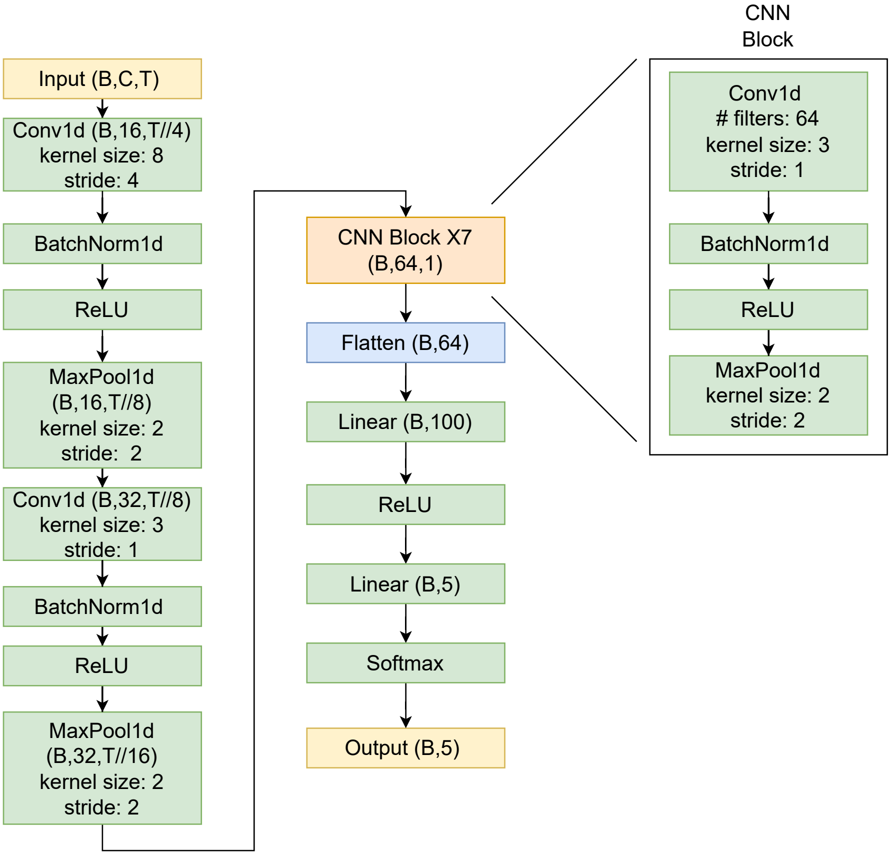

Input (B,C,T)
Conv1d (B,16,T//4) kernel size: 8 stride: 4
BatchNorm1d
ReLU
MaxPool1d (B,16,T//8) kernel size: 2 stride: 2
Conv1d (B,32,T//8) kernel size: 3 stride: 1
BatchNorm1d
ReLU
MaxPool1d (B,32,T//16) kernel size: 2 stride: 2
CNN Block X7 (B,64,1)
Flatten (B,64)
Linear (B,100)
ReLU
Linear (B,5)
Softmax
Output (B,5)
CNN Block
Conv1d # filters: 64 kernel size: 3 stride: 1
BatchNorm1d
ReLU
MaxPool1d kernel size: 2 stride: 2

Figure 3.8: Architecture of the CNN model from the Satapathy and Loganathan [39] paper. The figure contains the names and the output shape of the different layers. B represents the batch size, C the number of channels, and T the number of data points.

The architecture of the model is shown in Figure 3.8. The feature extraction part consists of nine convolution blocks. Each block includes a 1D-convolutional layer, a batch normalization layer, and a ReLU activation layer. The convolutional layers in each block have different filter sizes to capture different information. Each block is followed by a max-pooling layer to reduce the spatial dimensions of the features. The classifier is composed of 2 dense layers, with the first layer followed by a ReLU activation function and the second layer by a Softmax activation function. The model has a total of 90,285 parameters. For training, we employed the weighted cross-entropy loss and the Adam optimizer algorithm.

3.2.5 Metrics

Metrics are an important topic in machine learning as they are directly related to the goals we are trying to achieve or the hypotheses we are testing. Different metrics are often reported, but there is a need for a main metric that will allow us to compare different systems, while secondary metrics are reported to facilitate comparisons with the existing literature.

Table 3.5: Number and percentage for each class of labels and for each dataset

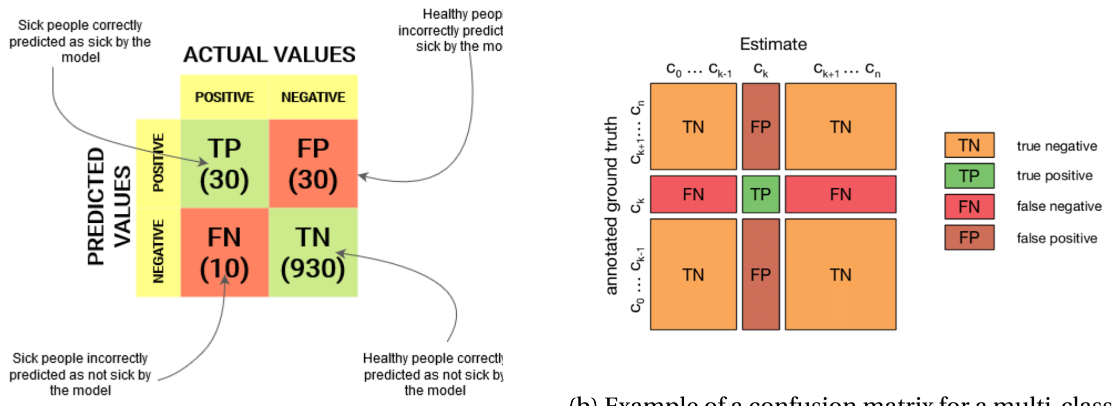
Dataset	Wake	Stage 1	Stage 2	Stage 3	REM	Total
SC-EDF (cropped)	66232 (34%)	21522 (11%)	69132 (35%)	13039 (7%)	25835 (13%)	195760
ST-EDF	4506 (11%)	3653 (9%)	19851 (46%)	13039 (15%)	25835 (19%)	42774
SS3-MASS	6442 (11%)	4839 (8%)	29802 (50%)	7653 (13%)	10581 (18%)	59317

The machine learning problem we are dealing with is an imbalanced problem due to the nature of sleep. Specifically, the N2 phase is more prevalent than the others in a healthy person as shown in Table 3.5. This characteristic has led us to choose the balanced accuracy as the main metric. By using this metric, we make the assumption that it will prevent the model from overfitting to the most frequent class and thus enable better generalization. Additionally, we report accuracy and Cohen's Kappa to facilitate comparisons with the existing literature.

Before defining all these metrics, let's first define some terms related to a binary classification problem. We will then extend them to a multi-class classification problem [2]:

[2] Refer to Grandini, Bagli, and Visani [62] for examples and more detailed explanations on multi-class metrics

- **True Positive (TP)**: The output is a TP if the predicted value and the true value are both positive.
- **True Negative (TN)**: The output is a TN if the predicted value and the true value are both negative.
- **False Positive (FP)**: The output is an FP if the predicted value is positive, but the true value is actually negative.
- **False Negative (FN)**: The output is an FN if the predicted value is negative, but the true value is actually positive.



(a) Example of a confusion matrix for a binary classification problem. (Source [63])

(b) Example of a confusion matrix for a multi-class classification problem. (Source [64])

Figure 3.9: Figure 3.9a shows an example of a confusion matrix for a binary classification problem, where the task is to estimate if an individual is sick or not. Figure 3.9b shows the distribution of TP, TN, FP, and FN for the c_k class in a multi-class classification problem.

An example of binary classification problem outputs can be seen in Figure 3.9a. To extend TP, TN, FP, and FN to multi-class, we use the one-vs-all classification solution. This means computing N binary classifications for an N-class classification problem. Figure 3.9b shows an example of this process for the c_k class, considering c_k as positive and all other classes as negatives. By following this principle for all classes, we build the multi-class confusion matrix.

Accuracy

The accuracy of a model is a measure of its ability to make correct predictions. For a confusion matrix M of an N-class classification problem, the accuracy is defined by Equation 3.2.

$$\text{Accuracy} = \frac{\sum_i^N TP_i}{t} \tag{3.2}$$

where:

- $t = \sum_i^N \sum_j^N M_{ij}$, the total number of elements in the matrix M.

Balanced Accuracy

The balanced accuracy for a multi-class problem is simply the average of the recall of each class, which gives Equation 3.3 for an N-class classification problem. The recall is the accuracy score for each class respectively. From the confusion matrix, it means to take, for each class, the diagonal element (TP) and divide it by the sum of all elements in the row.

$$\text{Balanced Accuracy} = \frac{\sum_i^N \frac{TP_i}{TP_i + FN_i}}{N} \tag{3.3}$$

Cohen's Kappa

Cohen's Kappa is a metric used to compute the inter-rater agreement between two scorers, as explained in Section 2.2. For a confusion matrix M of an N-class classification problem, Cohen's Kappa follows Equation 3.4.

$$\text{Cohen's Kappa} = \frac{\sum_{i=1}^N TP_i \times t - \sum_c^N \sum_i^N M_{ci} \times \sum_i^N M_{ic}}{t^2 - \sum_c^N \sum_i^N M_{ci} \times \sum_i^N M_{ic}} \tag{3.4}$$

where:

- $t = \sum_i^N \sum_j^N M_{ij}$, the total number of elements in the matrix M.
- $\sum_i^N M_{ci}$ is the sum of predictions for the class c (sum of column elements).
- $\sum_i^N M_{ic}$ is the sum of positive true cases for the class c (sum of row elements).

3.2.6 Evaluation Protocol

Following our goal of testing the generalization capacity across datasets of our models,we developed three different protocols described in Table 3.6, where for each one, one dataset is used for training, and the other ones are used for cross-dataset analysis.

Each dataset is composed of three subsets (train/validation/test). while running, we reported the balanced accuracy (our metric) for each subset of each dataset to perform analysis. However, to compare the different models in our experiments, we need a way to aggregate the results of each subset to get a unique score.

To achieve this, we designed the evaluation protocol as shown in Figure 3.10. As we are interested in cross-dataset results, we aggregated only the datasets that have not been used in

Table 3.6: Dataset composition of the different protocols and what they have been used for (train/validation or test) as well as the preprocessing techniques which have been applied to each dataset.

Protocol	ST-EDF (Sleep-EDF)	SC-EDF (Sleep-EDF)	SS3 (MASS)
EDF-MASSA	Train[a]	Test[a]	Test[b]
EDF-MASSD	Test[a]	Test[a]	Train[b]
EDF-MASSE	Test[a]	Train[c]	Test[b]

[a] Filtering, Label Fusion
[b] Filtering, Channel Fusion, Bipolar Reference Computation, Re-sampling
[c] Filtering, Label Fusion, Crop Wake Time

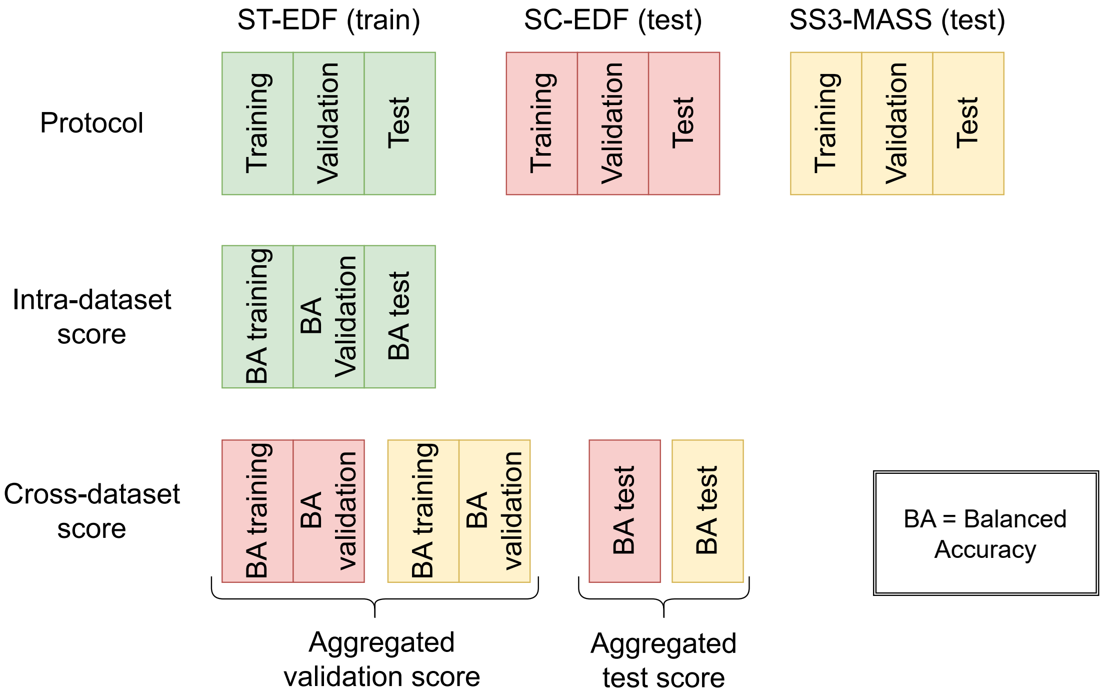


Figure 3.10: Evaluation protocol for EDF-MASSA protocol

the training process. We combined the training and validation subsets to compute a validation cross-dataset balanced accuracy score, and the test subsets were used to calculate a test cross-dataset balanced accuracy score.

$$\text{Weighted average} = \frac{\sum_i \text{Balanced Accuracy}_i \times w_i}{\sum_i w_i} \tag{3.5}$$

where:

- w_i is the number of 30s-windows in the i^{th} subset.
- Balanced Accuracy$_i$ is the balanced accuracy score of the i^{th} subset.

To aggregate the balanced accuracy of the different subsets, we used the weighted average described by Equation 3.5.

While performing an experiment, the best model will be the one with the highest validation cross-dataset balanced accuracy score, and the test cross-dataset balanced accuracy score will be reported.

3.2.7 Fusion of multi-modal data

Multi-modal fusion is a technique used within machine learning to combine information from different modalities or sources. There are different ways of fusing multivariate or multi-modal data, and this fusion can be performed at various stages in the machine learning pipeline. The objective of multi-modal fusion is to create a more comprehensive representation by leveraging complementary information available from various sources

Figure 3.11 illustrates the working flow of the three fusion methods utilized in this study: early, mid, and late fusion. Early fusion is a data-level fusion method that involves stacking the input data together. Mid fusion, on the other hand, is a feature fusion method where the feature vectors are concatenated to combine the information. Lastly, late fusion is a decision fusion method, which is achieved by performing a weighted sum of the predictions from each classifier.

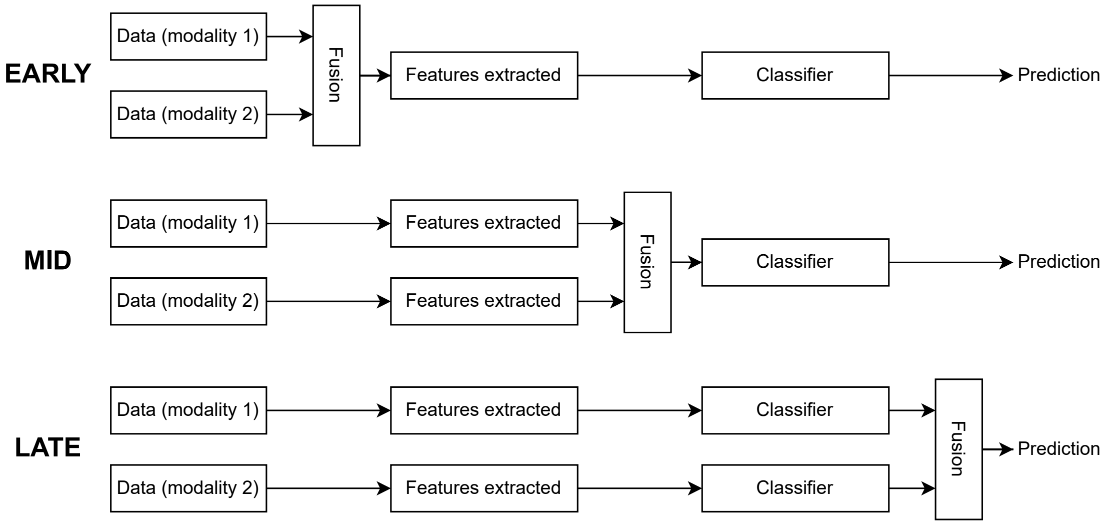


Figure 3.11: Different methods of fusion for multi-modality data. Pipeline 1 fuses the raw data, the second pipeline fuses the extracted features, and the last pipeline fuses the classification predictions. (Inspired from [65])

4 Experiments

Throughout this work, we conducted several experiments on our models, and in this section, we present eight of these experiments. Table A.2 provides the naming convention and command line for reproducing the experiments with the sleepless package using the hand-crafted feature extraction model, while Table A.1 provides the same information for the CNN model.

4.1 Experimental Setup

4.1.1 Channels

As explained in the previous Section 3, initially, we decided to work with similar channels. Since the SC-EDF and ST-EDF datasets share two channels (Fpz-Cz and Pz-Oz), we computed these two channels for the SS3-MASS dataset. We achieved this by interpolating Fpz with Fp1 and Fp2. Subsequently, we calculated a bipolar reference between the interpolated channel Fpz-LER and Cz-LER to get Fpz-Cz, as well as between Pz-LER and Oz-LER to obtain Pz-Oz.

A similar operation was performed for the other modalities (EOG, EMG) of the SS3-MASS dataset. Computation of bipolar reference for the EOG and average combination of channels for the EMG.

4.1.2 Grid-Search

For classic machine learning algorithms like random forest and gradient boosting, we performed hyperparameter selection using grid search.

In the case of the random forest model, we optimized the following hyperparameters:

- Number of estimators: [100, 200, 300]
- Maximum depth: [4, 6, 8, 10]

- Criterion: ['gini', 'entropy', 'log_loss']
- Bootstrap: [True, False]
- Class weight: ['balanced', None]

Similarly, for the gradient boosting model, we selected the following hyperparameters:

- Learning rate: [10e-4, 10e-3, 10e-2]
- Minimum child weight: [2, 4, 6, 8, 10]
- Maximum depth: [2, 4, 6, 8, 10]
- Alpha: [0, 0.5, 1]
- Column sampling by level: [0.5, 0.75, 1]

By systematically evaluating the performance of the models with different combinations of these hyperparameters through grid search, we aim to identify the optimal set of hyperparameters that maximize the models' balanced accuracy. This process was performed with the training and validation intra-dataset.

In the case of a neural network model, the validation set is used to choose the optimal model and assess the quality of the training process through learning curves.

4.2 Experiment 1: Gradient boosting versus random forest

In this first experiment, we decided to compute the results of our MNE baseline described in Section 3.2.4 with 2 EEG channels Fpz-Cz and Pz-Oz. The method used to fuse information is the mid fusion technique described in Section 3.2.7. Additionally, we replaced the random forest with a gradient boosting model for comparison.

The objective of this experiment was to assess and compare the classification performance of the gradient boosting model against the random forest model. We anticipated that gradient boosting, known for its ability to mitigate overfitting and enhance generalization, would outperform the random forest algorithm.

For each experiment, we conducted an intra-database analysis of each model to assess the quality of the training process. As an example, we analyzed the random forest model trained on the SC-EDF dataset (EDF-MASS-E). Through a grid search, we identified the following chosen parameters: number of estimators: 100, maximum depth: 10, criterion: entropy, bootstrap: True, class weight: balanced.

Subset	Train	Validation	Test
Balanced accuracy	0.72	0.73	0.68

Table 4.1: Experiment 1: Intra-dataset balanced accuracy score for the random forest model trained on the SC-EDF dataset.

Considering our evaluation metric, which is the balanced accuracy, it is appropriate to set the class weight parameter to 'balanced' due to the dataset's imbalance. Furthermore, since the maximum depth parameter reached the highest possible value, we needed to verify that the model did not suffer from overfitting. In the random forest model, a high value of maximum depth often indicates a potential overfitting issue.

Table 4.1 displays the intra-dataset balanced accuracy scores. We observe that the validation set score is slightly higher than the training score, which confirms that the model did not overfit. This observation is further supported by the test set score, which is also very close to the training and validation scores.

Figure 4.1 illustrates the results of our experiment, for our three protocols. We can observe that our hypothesis is not confirmed as for the three protocols the random forest model gets a better validation score.

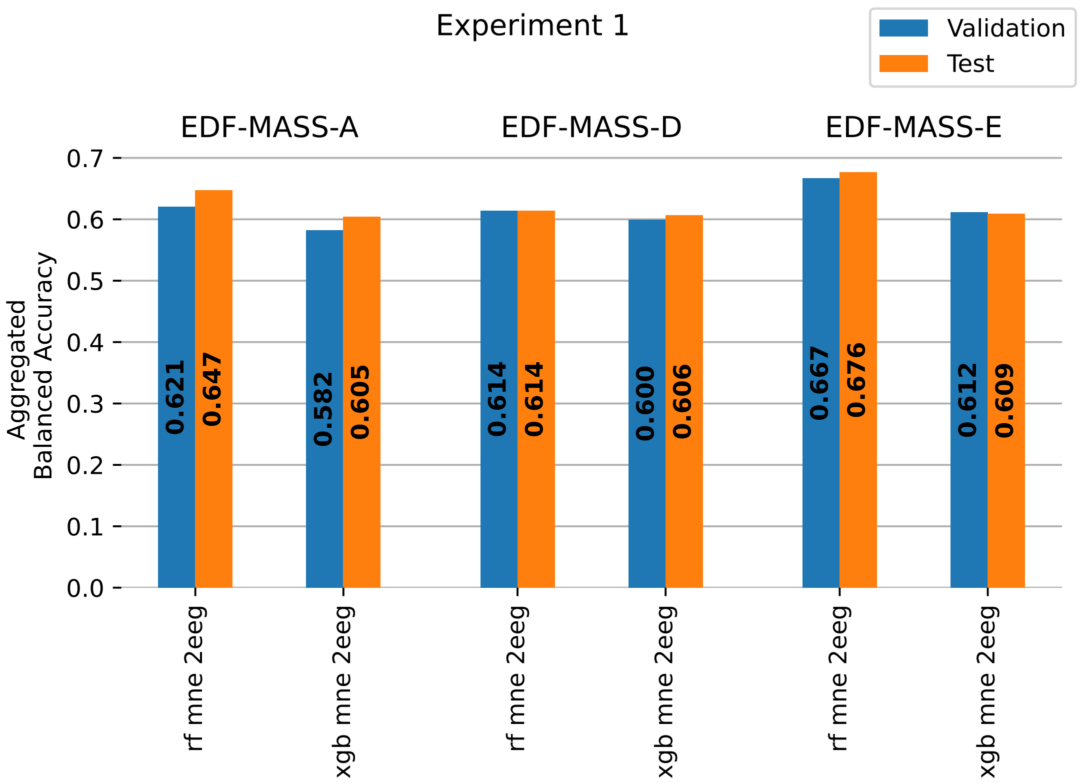


Figure 4.1: Validation and test aggregated balanced accuracy for random forest and gradient boosting model with the MNE baseline feature extraction algorithm. The results were computed for 3 protocols EDF-MASS-A, EDF-MASS-D, and EDF-MASS-E where the models were trained respectively on ST-EDF, SC-EDF, and SS3-MASS datasets. Two EEG channels, specifically Fpz-Cz and Pz-Oz, were used as input for the models.

4.3 Experiment 2: What is the effect of increasing the number of features?

To analyze the effect of increasing the number of features, we compared three models. The first model was a random forest with 5 features extracted (MNE baseline) per channel, which was determined as the best model in the previous experiment. The second model was a gradient boosting model with 26 features extracted (Chambon hand-Craft features baseline) per channel. Additionally, based on the results of the previous experiments, we implemented a random forest model using the features extracted from the Chambon hand-Craft features baseline. All of these results were computed using 2 EEG channels.

Subset	Train	Validation	Test
Balanced accuracy	0.84	0.85	0.68
Chambon et al. [1] Balanced accuracy	N/A	N/A	0.7

Table 4.2: Experiment 2: Intra-dataset balanced accuracy score for the gradient boosting model trained on the ST-EDF dataset.

If we examine the intra-database results of the gradient boosting model with the Chambon feature extractor algorithm presented in Table 4.2 for the EDF-MASS-A protocol, we can observe the training quality of the model. This is evident from the close proximity of the training and validation scores. However, there may be slight overfitting, as indicated by the performance gap between the validation and test sets.

In their paper, Chambon et al. [1] achieved a balanced accuracy of approximately 70% for 2 EEG channels on their test set using gradient boosting. This result is comparable to our obtained balanced accuracy of 68% for the EDF-MASS-A protocol which validates our implementation.

From Figure 4.2, we can draw two conclusions. Firstly, it confirms the findings of the first experiment that the random forest model outperforms the gradient boosting model for all three protocols. Secondly, we observed that the Chambon feature extraction algorithm achieved higher validation aggregated balanced accuracy scores for all protocols. This result was expected because the Chambon features include a wider range of information, such as frequency, temporal, and entropy, whereas the MNE features are limited to frequency information only. Therefore, the Chambon features bring complementary and wider information, leading to improved performance.

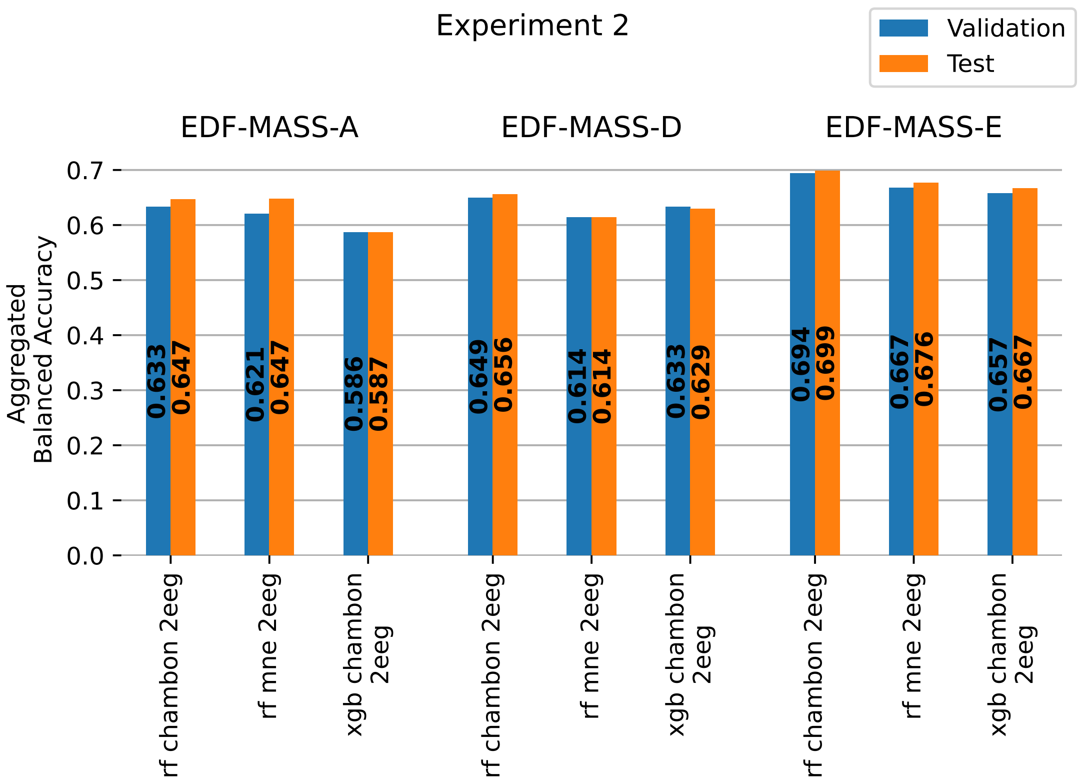


Figure 4.2: Validation and test aggregated balanced accuracy for random forest, gradient boosting model with the Chambon baseline feature extraction algorithm and for random forest model with the MNE baseline feature extraction algorithm. The results were computed for 3 protocols EDF-MASS-A, EDF-MASS-D, and EDF-MASS-E where the models were trained respectively on ST-EDF, SC-EDF, and SS3-MASS datasets. Two EEG channels, specifically Fpz-Cz and Pz-Oz, were used as input for the models.

4.4 Experiment 3: What is the influence of the location and the number of EEG channels?

To conduct this experiment, we selected the best algorithm from previous experiments, which is the random forest with the Chambon baseline feature extraction method using 2 EEG channels. Additionally, we computed the results for the same model and feature extraction algorithm, but for each channel separately: Fpz-Cz and Pz-Oz.

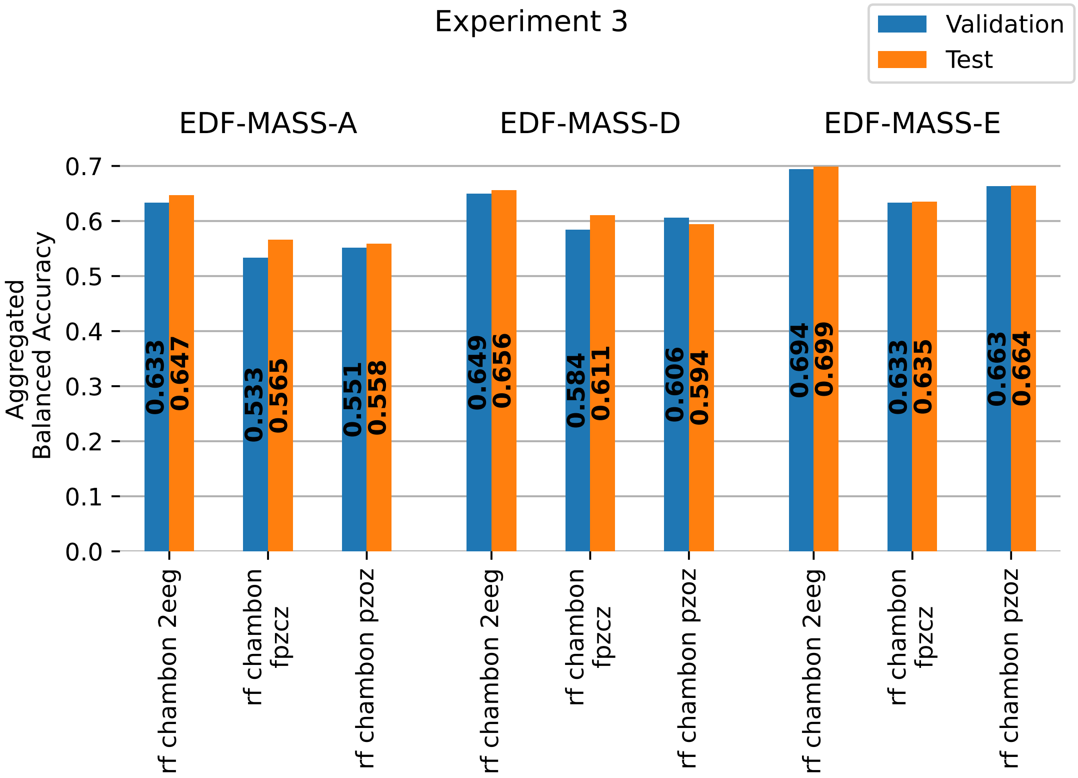


Figure 4.3: Validation and test aggregated balanced accuracy for random forest with the Chambon baseline feature extraction algorithm. The results were computed for 3 protocols EDF-MASS-A, EDF-MASS-D, EDF-MASS-E where the models were trained respectively on ST-EDF, SC-EDF and SS3-MASS datasets. Three models were trained: one on both Fpz-Cz and Pz-Oz channels, one on Fpz-Cz channel only, and one on Pz-Oz channel only.

The Figure 4.3 demonstrates that the model computed with 2 EEG channels as input outperforms the models using individual channels for all three protocols. It is evident that not all channels contribute equally to the classification of sleep stages. Specifically, the Pz-Cz channel seems to contain more relevant information for sleep stage classification compared to the Fpz-Cz channel. However, the utilization of both channels appears to provide complementary information, resulting in an increase ranging between 3 and 7% in the validation aggregated balanced accuracy for the model incorporating both channels.

4.5 Experiment 4: Which modality brings the more information?

For this experiment, our aim was to analyze the classification scores obtained for each individual modality. We specifically selected EEG, EOG, and EMG as the modalities for this study. we employed the same Chambon feature extraction algorithm along with a random forest model for all modalities.

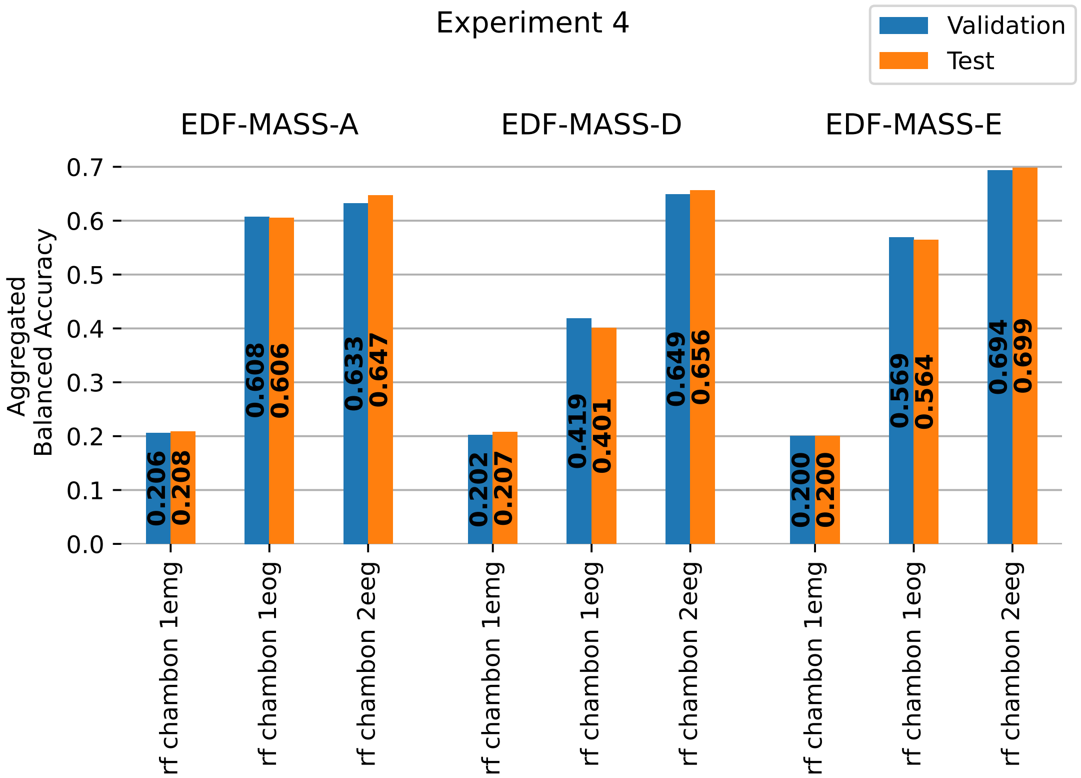


Figure 4.4: Validation and test aggregated balanced accuracy for random forest with the Chambon baseline feature extraction algorithm. The results were computed for 3 protocols EDF-MASS-A, EDF-MASS-D, EDF-MASS-E where the models were trained respectively on ST-EDF, SC-EDF and SS3-MASS datasets. Three models were trained: one on both Fpz-Cz and Pz-Oz channels, one on the EOG channel, and one on the EMG channel.

Figure 4.4 demonstrates that the models trained on the EEG channels exhibit higher accuracy across all three protocols. The models trained on the EOG modality ranked second and achieved nearly 60% aggregated balanced accuracy, except for the EDF-MASS-D protocol. It appears that the EDF-MASS-D protocol has lower generalization capabilities compared to the other protocols for this particular modality. Additionally, it is worth noting that the EMG modality did not exhibit any generalization capacity, as indicated by its aggregated balanced accuracy of 20% across all protocols.

4.6 Experiment 5: Is mid fusion of modalities improve performance?

We decided to analyze the impact of fusing modalities using a mid fusion method, specifically by concatenating the features extracted by the feature extractor algorithm. Building upon the previous experiment, we employed a random forest model with a Chambon feature extractor algorithm. We performed the following fusion combinations: EEG + EOG, EEG + EMG, EEG + EOG + EMG. We compared the results of these fusion combinations with the model that solely utilized EEG, which was the best-performing model we obtained thus far.

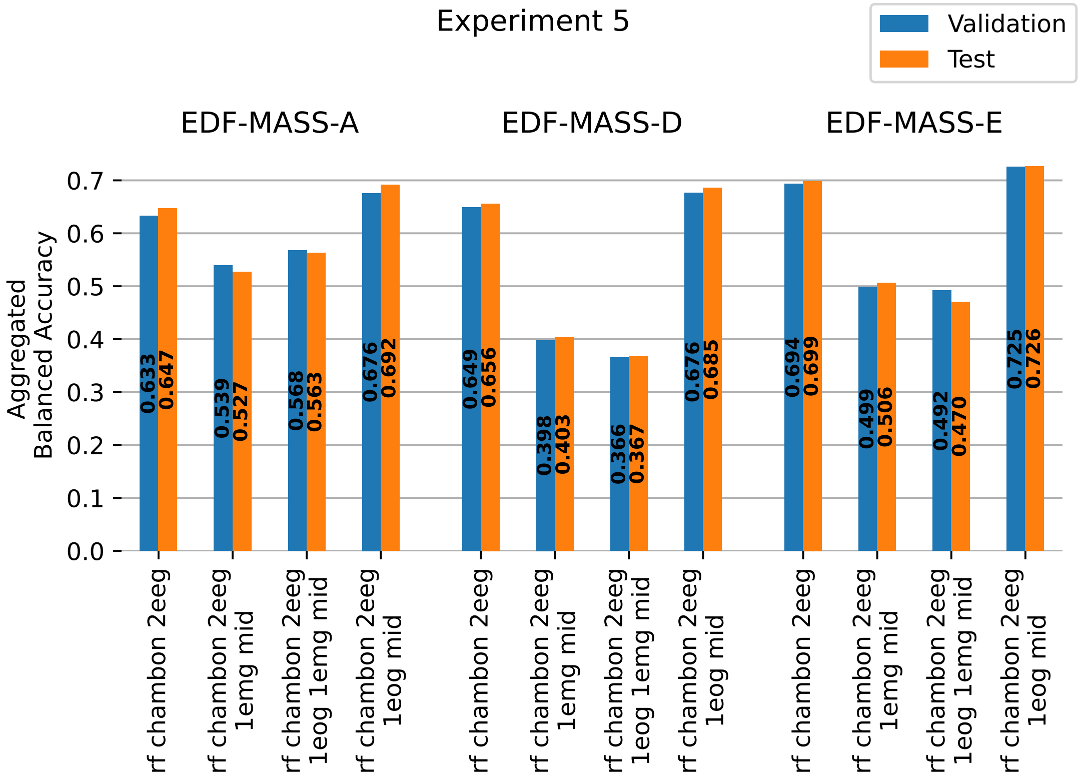


Figure 4.5: The aggregated balanced accuracy for validation and test sets was computed using the random forest algorithm with the Chambon baseline feature extraction algorithm. The results were obtained for three protocols: EDF-MASS-A, EDF-MASS-D, and EDF-MASS-E. The models were trained on the ST-EDF, SC-EDF, and SS3-MASS datasets, respectively. Four models were trained: one using both Fpz-Cz and Pz-Oz channels, one using a mid fusion of EEG + EOG channels, one using a mid fusion of EEG + EMG channels, and one using a mid fusion of EEG + EMG + EOG channels.

Figure 4.5 illustrates that, across all three protocols, the mid fusion combination of EEG + EOG demonstrates a higher aggregated balanced accuracy compared to using EEG alone. Furthermore, the inclusion of the EMG modality reduces the generalization capacity of the model. This observation aligns with the findings from Experiment 5, where the EMG modality exhibited no generalization capacity and instead introduced noisy information for the classifier. Hence, it can be inferred that incorporating the EMG modality does not contribute beneficial information for classification and may even degrade the performance of the model.

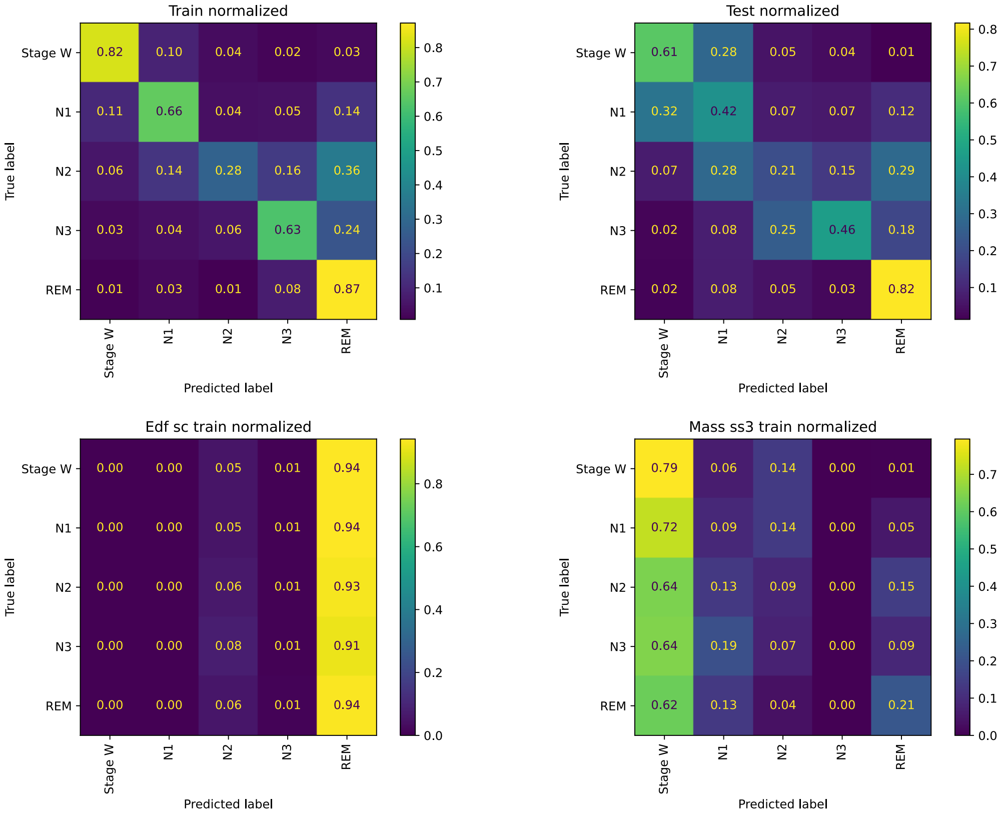


Figure 4.6: The different confusion matrices illustrate the classification results of a random forest model that solely uses the EMG channel. The top section displays the outcomes for the train (left) and test (right) subsets of the ST-EDF dataset, while the bottom section shows the cross-dataset confusion matrices for the train subsets of the EDF-SC (left) and MASS-SS3 (right) datasets.

One conceivable hypothesis to account for the suboptimal outcomes observed within the EMG modality pertains to a deficiency in generalization capability. This phenomenon is notably exemplified through the visualization presented in Figure 4.6. The depicted confusion matrices are a product of employing a random forest model exclusively trained on the EMG channel, and they serve as a representation of this observed trend. While these results are anchored in the context of the EDF-MASS-A protocol, comparable trends manifest across diverse protocols.

By closely examining the depicted figure and scrutinizing the train and test subsets, we discern that the model has the capacity to learn patterns from the EMG channel. The outcomes in the test subsets, although not optimal (some sleep stages under 50% of accuracy), manifest the model's ability to effectively classify certain sleep stages, notably stages W (Wake) and REM (Rapid Eye Movement), albeit with a degree of limited precision.

Upon meticulous examination of the lowermost confusion matrices presented in Figure 4.6, specifically those pertaining to the edf sc and masss ss3 train subsets, a pronounced deficiency in generalization becomes conspicuously manifest. It is crucial to emphasize that the model under consideration was exclusively trained on edf st datasets.

The deficiency in generalization could potentially be attributed to the inherent nature of utilizing EMG signals for sleep stage classification. The classification process reliant on EMG involves interpreting the absolute level of muscle tonus, which could inherently vary across different experimental setups and datasets.

4.7 Experiment 6: Is late fusion better than mid fusion?

In this experiment, we compared the best results from Experiment 5 (mid fusion) with a late fusion technique for the same combination of modality EEG + EOG. We performed the weighted sum of predictions as described in Equation 4.1, and conducted the experiment for 10 values of α ranging between 0 and 1, evenly spaced.

$$\text{Late Fusion Prediction} = \alpha \cdot Prediction_{EEG} + (1 - \alpha) \cdot Prediction_{EOG} \tag{4.1}$$

Where:

- $Prediction_{EEG}$ are the predictions from the model rf chambon 2eeg
- $Prediction_{EOG}$ are the predictions from rf chambon 1eog

Figure 4.7 illustrates that the mid fusion technique achieved a higher score than the late fusion method. Across all three protocols, increasing the parameter α, which gives more weight to the classifier trained solely on EEG, improved the performance until an alpha value of 0.6. However, beyond an alpha of 0.6, the performance started to decrease.

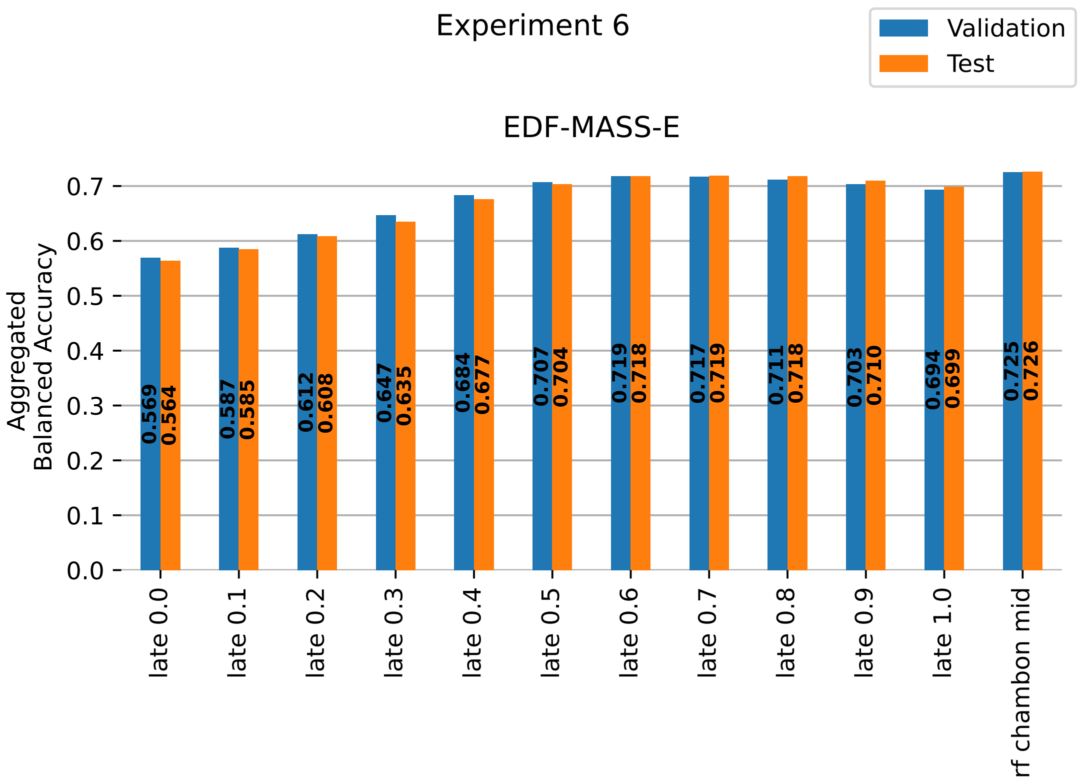


Figure 4.7: The validation and test aggregated balanced accuracy for the random forest algorithm with the Chambon baseline feature extraction algorithm was computed for the protocol EDF-MASS-E. The models were trained on the SC-EDF dataset. One model was trained using a mid fusion of EEG + EOG channels. Additionally, a weighting sum (late fusion) approach was employed to combine the predictions of a model trained solely on EEG and a model trained solely on EOG, using different alpha values. The parameter alpha determines the weight given to the model trained on EEG channels, with increasing alpha resulting in a higher weight for the EEG model in the fusion process.

4.8 Experiment 7: Is a learnable feature extraction model better than a manually chosen feature algorithm?

We subsequently opted to explore models with a learnable feature extraction layer, such as CNN, and compare their results with the random forest model using the Chambon feature extractor algorithm. For this experiment, we computed the results for the Chambon CNN model, described in Section 3.2.4. Building upon the previous experiments, we trained the CNN model using both EEG and EOG channels, employing an early fusion method (signal stacking) to combine the input data.

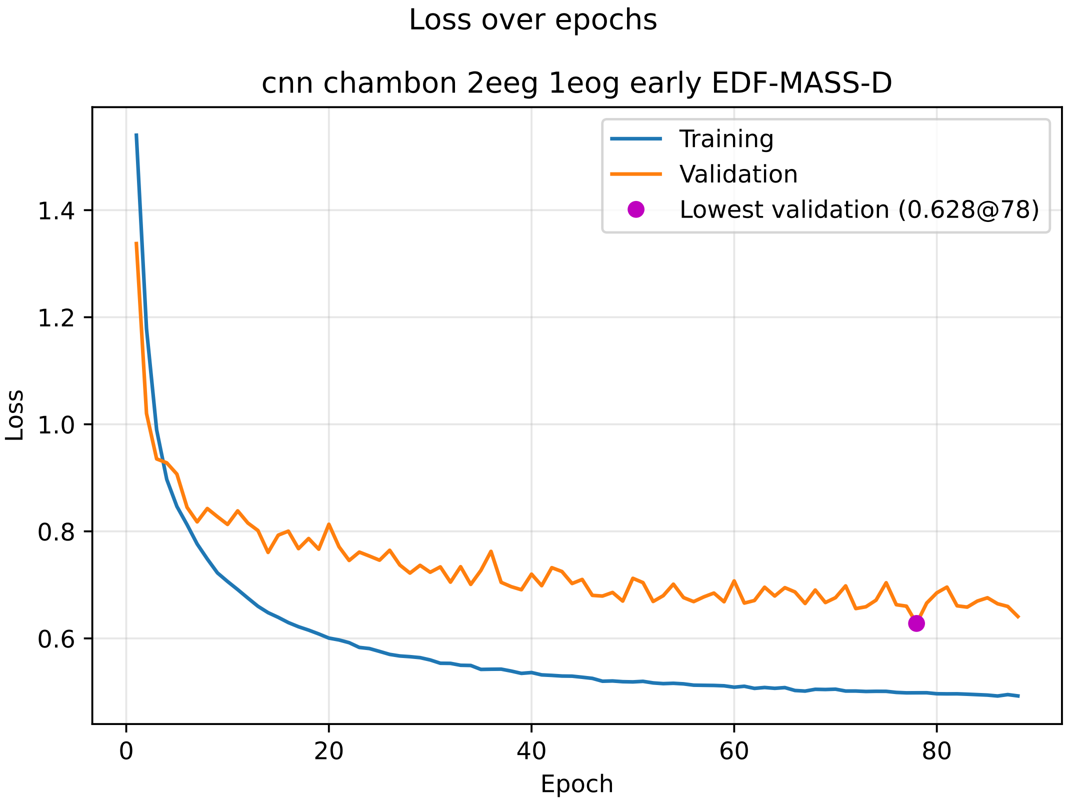


Figure 4.8: Learning curves over epochs for the Chambon CNN model trained and validated on the SS3-MASS dataset.

Figure 4.8 displays the learning curves of the CNN model trained on the EDF-MASS-D protocol. The figure indicates that both the validation and training curves have reached a plateau, suggesting that the model has converged. However, there is a noticeable gap between the training and validation curves, which typically indicates overfitting. It is important to note that since the validation curve does not increase further towards the end of training, it suggests that the overfitting is minimal.

In their paper, Chambon et al. [1] achieved a balanced accuracy of 80% on their test set using

Subset	Train	Validation	Test
Our Balanced accuracy	0.82	0.75	0.79
Chambon et al. [1] Balanced accuracy	N/A	N/A	0.8

Table 4.3: Experiment 7: Intra-dataset balanced accuracy score for the Chambon CNN model trained on the SS3-MASS dataset.

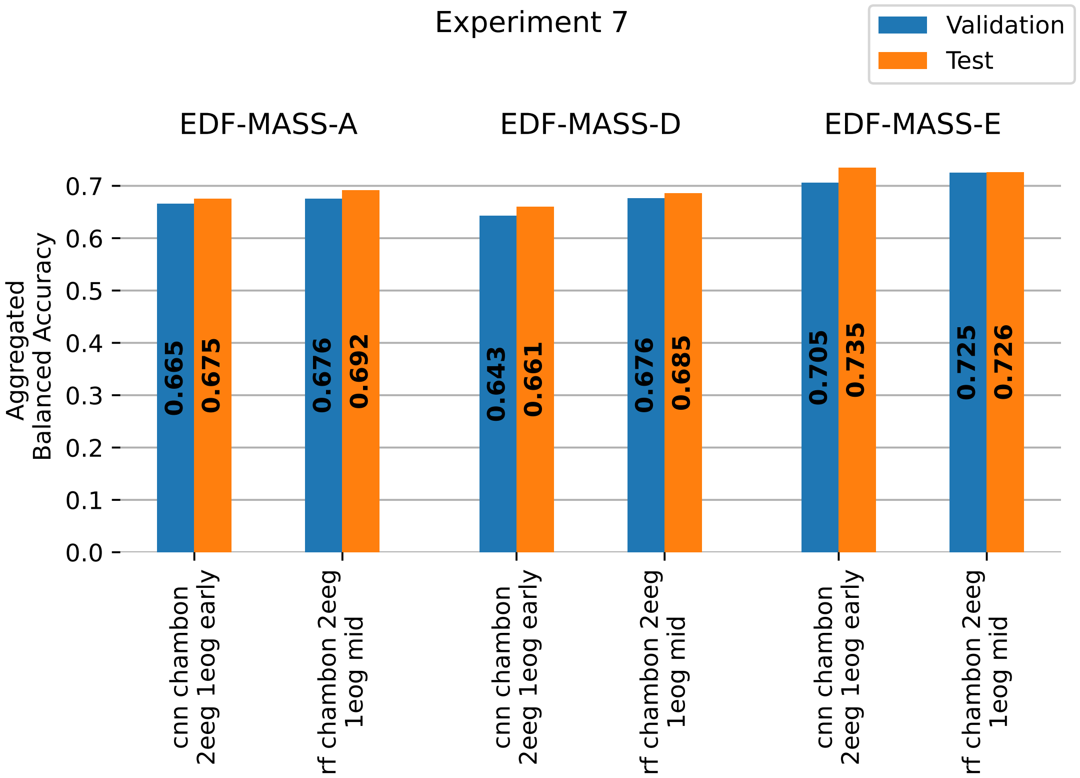


Figure 4.9: The validation and test aggregated balanced accuracy were computed for the random forest model utilizing the Chambon baseline feature extraction algorithm and for the Chambon CNN model for three protocols: EDF-MASS-A, EDF-MASS-D, and EDF-MASS-E. The models were trained on the ST-EDF, SC-EDF, and SS3-MASS datasets, respectively. Both models were trained using a combination of EEG + EOG channels. However, the CNN model employed an early fusion method, while the random forest model utilized a mid fusion technique.

their CNN model, which was trained on a combination of EEG and EOF channels. Table 4.3 presents the results of our CNN model trained on the SS3-MASS dataset. It can be observed that our model obtained a similar balanced accuracy score of 79% on the test set.

Figure 4.9 demonstrates that the aggregated balanced accuracy of the CNN models is slightly lower compared to the random forest model using hand-crafted features. Although the difference is relatively small, ranging from 1-4%, the random forest with Chambon feature extractor algorithm is still the better model for each protocol.

4.9 Experiment 8: Does an increase in the number of parameters increases the balanced accuracy?

In this experiment, we computed the results for the model from the paper by Satapathy and Loganathan [39]. The aim of this experiment is to analyze the effect of increasing the number of parameters on generalization. Specifically, the Chambon CNN model used in our study has only 6,033 parameters, whereas the nine-layer CNN described in Section 3.2.4 has 90,285 parameters.

Subset	Train	Validation	Test
Our Accuracy	0.77	0.74	0.71
Satapathy and Loganathan [39] Accuracy	0.98	N/A	0.98

Table 4.4: Experiment 8: Intra-dataset accuracy score for the Satapathy CNN model trained on the SC-EDF dataset.

Looking at Table 4.4, we can compare our accuracy score with the score obtained in the paper [39]. On their test set, they achieved an accuracy of 98% with only one EEG channel, whereas we obtained an accuracy of 71% with a combination of EEG+EOG channels on our test set. Based on these results, we could estimate that the model was underfit and could be trained further. However, examining the learning curve in Figure 4.10, we observe that while the training curves continue to decrease at the end of training, the validation curves have already reached a plateau. This indicates that further training would overfit the model.

Another reason to explain the difference in results is that our protocol did not contain enough samples to train a model with that many parameters. However, in [39], they used only 2,625 30-second windows to train their model, whereas we used 122,278 30-second windows for training. We were then unable to reproduce comparable results.

Figure 4.11 demonstrates that for our three protocols, increasing the number of parameters decreases the aggregated balanced accuracy score. The Chambon CNN model achieved a higher aggregated balanced accuracy, ranging between 4-5

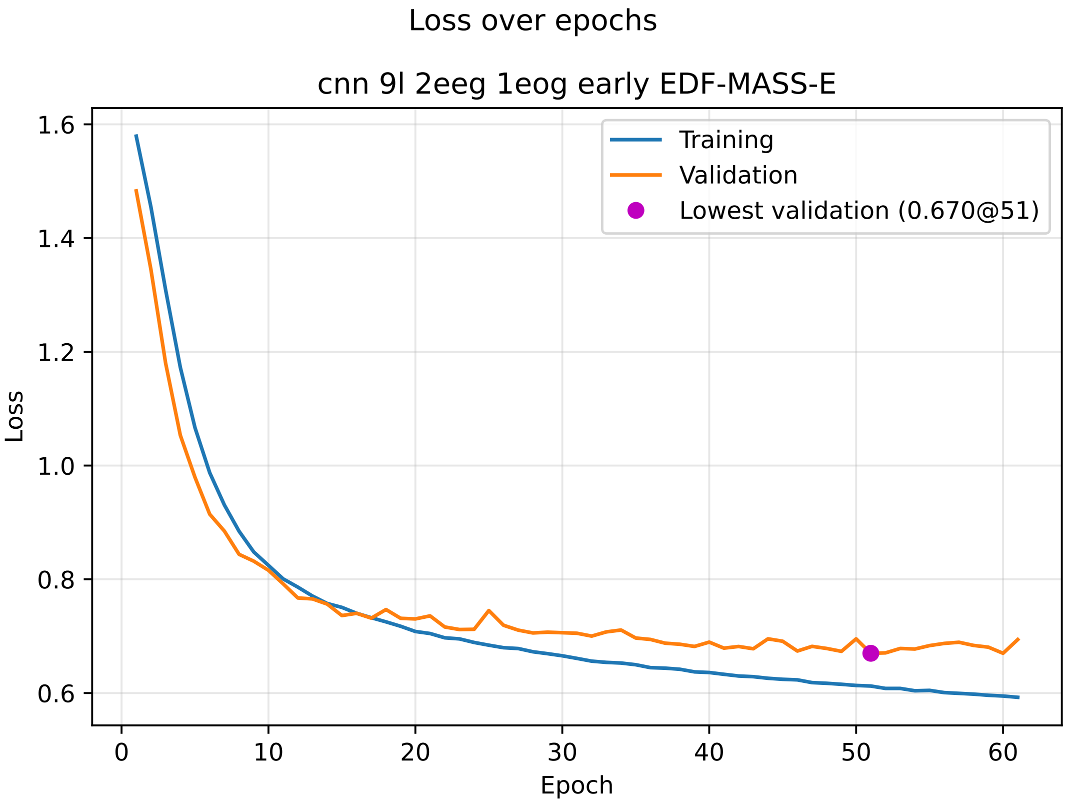


Figure 4.10: Learning curves over epochs for the Chambon CNN model trained and validated on the SC-EDF dataset.

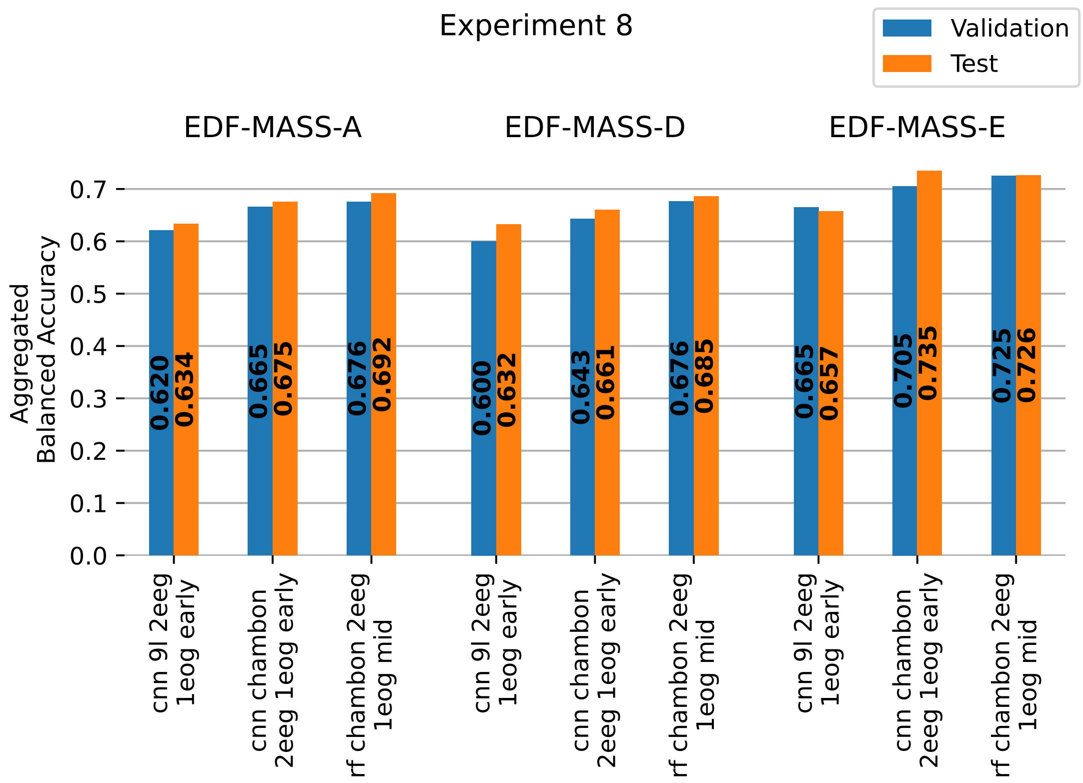


Figure 4.11: The validation and test aggregated balanced accuracy were computed for the random forest model utilizing the Chambon baseline CNN model and for the Satapathy CNN model for three protocols: EDF-MASS-A, EDF-MASS-D, and EDF-MASS-E. The models were trained on the ST-EDF, SC-EDF, and SS3-MASS datasets, respectively. Both models were trained using a combination of EEG + EOG channels. However, the CNN model employed an early fusion method, while the random forest model utilized a mid fusion technique.

Conclusion

From all our experiments, we can draw the conclusion that the random forest model with the Chambon hand-crafted feature extractor for a combination of EEG+EOG channels achieved the highest validation aggregated balanced accuracy. This indicates that this model has better generalization across different setups.

We initially expected that a learnable feature extractor model would outperform this baseline, as has been observed in various fields of machine learning. However, considering our specific experimental setups and evaluation protocol, we did not surpass the performance of the random forest model with hand-crafted features.

From the results of these experiments, we compared the performance of our best model with the human inter-rater Cohen's Kappa. Our model achieved a Cohen's Kappa on the intra-database test set ranging from 0.67 to 0.72, depending on the protocols. Our model did not surpass the inter-rater score of humans, which was reported to be a Cohen's Kappa of 0.76 [29].

The failure to outperform the random forest model with CNN models and to surpass the inter-rater score of humans may be attributed to the evaluation protocols we designed and the metrics we decided to work with. Investigating these two aspects could provide valuable insights for further advancement of this work.

Further investigation can be conducted in the field of stateful algorithms. Addressing the intricate classification of sleep stage 1, tied to transition rules, presents a promising avenue for enhancement. Similarly, refining the classification of sleep stage 2, which is notably intertwined with the sleep stage of preceding windows-epochs, could likely be improved by employing models that incorporate a memory system.

Future works could also conduct experiments on the algorithm's fairness concerning health, gender, and age. Investigating this aspect was one of our primary research questions, and we have developed the necessary tools to analyze the outcomes. However, due to constraints, we were unable to carry out experiments to empirically test our hypotheses pertaining to this inquiry.

A An appendix

Table A.1: Explanation of the naming convention used in this work and the respective comm-mand line to reproduce the experiment in the sleepless package

Name	Model	Channels	Command Line[a]
cnn chambon 2eeg 1eog	CNN[b]	Fpz-Cz, Pz-Oz, horizontal	chambon-noscheduler
cnn 9l 2eeg 1eog	CNN[c]	Fpz-Cz, Pz-Oz, horizontal	ninel-cnn

[a] model config file name to reproduce the experiment with the sleepless package

[b] Chambon-baseline see Section 3.2.4

[c] Satapathy-baseline see Section 3.2.4

Table A.2: Explanation of the naming convention used in this work and the respective commmand line to reproduce the experiment in the sleepless package for the manually chosen extraction techniques.

Name	Model	Feature Extraction	Channels	Command Line[a]
rf mne 2eeg	random forest	MNE[b]	Fpz-Cz and Pz-Oz	rf-gs-mne
xgb mne 2eeg	gradient boosting	MNE[b]	Fpz-Cz and Pz-Oz	xgb-gs-mne
rf chambon 2eeg	random forest	Chambon[c]	Fpz-Cz and Pz-Oz	rf-gs-chambon
xgb chambon 2eeg	gradient boosting	Chambon[c]	Fpz-Cz and Pz-Oz	xgb-gs-mne
rf chambon fpzcz	random forest	Chambon[c]	Fpz-Cz	rf-gs-chambon
rf chambon pzoz	random forest	Chambon[c]	Pz-Oz	rf-gs-chambon
rf chambon 1emg	random forest	Chambon[c]	Submental	rf-gs-chambon
rf chambon 1eog	random forest	Chambon[c]	Horizontal	rf-gs-chambon
rf chambon 2eeg 1emg	random forest	Chambon[c]	Fpz-Cz, Pz-Oz and submental	rf-gs-chambon
rf chambon 2eeg 1eog	random forest	Chambon[c]	Fpz-Cz, Pz-Oz and horizontal	rf-gs-chambon
rf chambon 2eeg 1eog 1emg	random forest	Chambon[c]	Fpz-Cz, Pz-Oz, horizontal and submental	rf-gs-chambon

[a] Model config file name to reproduce the experiment with the sleepless package
[b] MNE-baseline see Section 3.2.4
[c] Chambon-baseline hand-craft feature see Section 3.2.4

Table A.3: List of the most popular datasets in the literature.

Datasets	Number of PSG	Subjects	Accessibility
Mass [66]	200	healthy and patients	public [a]
MIT-BIH [67]	18	patients	public
CAP [68]	108	healthy and patients	public
Haaglanden Medisch Centrum [69]	151	healthy	public
Sleep Apnea [70]	25	patients	public
Sleep-EDF [71]	197	healthy	public
Dreams database [72]	47	healthy and patients	public
Siesta [73] [74]	295	healthy and patients	private
Massachusetts General Hospital Sleep Laboratory	10000	healthy	private
Sleep Heart Health Study [75] [76]	5804	patients	public
Psychiatry and Neurology in Warsaw	43	patients	private
Wisconsin Sleep Cohort	2310	healthy and patients	private
ISRUC-Sleep [77]	116	healthy and patients	public
University of Zurich [78]	54	healthy	private

[a] Need research ethic board approval

References

[1] Stanislas Chambon et al. “A Deep Learning Architecture for Temporal Sleep Stage Classification Using Multivariate and Multimodal Time Series”. en. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26.4 (Apr. 2018), pp. 758–769. ISSN: 1534-4320, 1558-0210. DOI: 10.1109/TNSRE.2018.2813138. URL: https://ieeexplore.ieee.org/document/8307462/ (visited on 05/19/2022).

[2] A. L. Loomis, E. N. Harvey, and G. A. Hobart. “Cerebral states during sleep, as studied by human brain potentials”. In: *Journal of Experimental Psychology* 21 (1937). Place: US Publisher: Psychological Review Company, pp. 127–144. ISSN: 0022-1015. DOI: 10.1037/h0057431.

[3] *Troubles du sommeil dans la population*. FR. 350824. Backup Publisher: Bundesamt für Statistik (BFS). Neuchâtel: Bundesamt für Statistik (BFS), May 2015. URL: https://dam-api.bfs.admin.ch/hub/api/dam/assets/350824/master.

[4] *5 Major Sleep Disorders*. en-US. Feb. 2020. URL: https://www.scinternalmedicine.com/2020/02/28/5-major-sleep-disorders/ (visited on 04/20/2022).

[5] David W. Carley and Sarah S. Farabi. “Physiology of Sleep”. en. In: *Diabetes Spectrum* 29.1 (Feb. 2016), pp. 5–9. ISSN: 1040-9165, 1944-7353. DOI: 10.2337/diaspect.29.1.5. URL: https://diabetesjournals.org/spectrum/article/29/1/5/32146/Physiology-of-Sleep (visited on 04/07/2022).

[6] *Sleep Stages: Understanding Your Sleep Cycles*. Nov. 2019. URL: https://casper.com/blog/ca/en/sleep-stages/ (visited on 04/21/2022).

[7] “Polysomnographic Assessment of DIMS: Empirical Evaluation of Its Diagnostic Value”. en. In: *Sleep* (July 1989). ISSN: 1550-9109. DOI: 10.1093/sleep/12.4.315. URL: https://academic.oup.com/sleep/article/12/4/315/2742672/Polysomnographic-Assessment-of-DIMS-Empirical (visited on 04/07/2022).

[8] Flavio Raschellà et al. *RBDAct: Home screening of REM sleep behaviour disorder based on wrist actigraphy in Parkinson’s patients*. en. preprint. Neurology, Jan. 2022. DOI: 10.1101/2022.01.23.22269713. URL: http://medrxiv.org/lookup/doi/10.1101/2022.01.23.22269713 (visited on 04/21/2022).

[9] Bradley V. Vaughn and Peterson Giallanza. “Technical Review of Polysomnography”. en. In: *Chest* 134.6 (Dec. 2008), pp. 1310–1319. ISSN: 00123692. DOI: 10.1378/chest.08-0812. URL: https://linkinghub.elsevier.com/retrieve/pii/S0012369209600349 (visited on 04/21/2022).

[10] Alexander Malafeev et al. “Automatic Human Sleep Stage Scoring Using Deep Neural Networks”. In: *Frontiers in Neuroscience* 12 (Nov. 2018), p. 781. ISSN: 1662-453X. DOI: 10.3389/fnins.2018.00781. URL: https://www.frontiersin.org/article/10.3389/fnins.2018.00781/full (visited on 02/15/2022).

[11] C. IBER. “The AASM Manual for the Scoring of Sleep and Associated Events : Rules”. In: *Terminology and Technical Specification* (2007). Publisher: American Academy of Sleep Medicine. URL: https://ci.nii.ac.jp/naid/10024500923/#cit (visited on 04/07/2022).

[12] A RECHTSCHAFFEN. “A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects”. In: *Brain information service* (1968). Publisher: Brain Research Institute. US Dept. of Health, Education, and Welfare. URL: https://ci.nii.ac.jp/naid/10027491188/ (visited on 04/07/2022).

[13] Heidi Danker-Hopfe et al. “Interrater reliability between scorers from eight European sleep laboratories in subjects with different sleep disorders”. en. In: *Journal of Sleep Research* 13.1 (2004). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1046/j.1365-2869.2003.00375.x, pp. 63–69. ISSN: 1365-2869. DOI: 10.1046/j.1365-2869.2003.00375.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-2869.2003.00375.x (visited on 04/08/2022).

[14] Luigi Fiorillo et al. “Automated sleep scoring: A review of the latest approaches”. en. In: *Sleep Medicine Reviews* 48 (Dec. 2019), p. 101204. ISSN: 10870792. DOI: 10.1016/j.smrv.2019.07.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S1087079218301746 (visited on 04/15/2022).

[15] Alexandra de Raadt et al. “A Comparison of Reliability Coefficients for Ordinal Rating Scales”. en. In: *Journal of Classification* 38.3 (Oct. 2021), pp. 519–543. ISSN: 0176-4268, 1432-1343. DOI: 10.1007/s00357-021-09386-5. URL: https://link.springer.com/10.1007/s00357-021-09386-5 (visited on 07/15/2022).

[16] Terry K. Koo and Mae Y. Li. “A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research”. en. In: *Journal of Chiropractic Medicine* 15.2 (June 2016), pp. 155–163. ISSN: 15563707. DOI: 10.1016/j.jcm.2016.02.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S1556370716000158 (visited on 07/15/2022).

[17] Magdy Younes et al. “Reliability of the American Academy of Sleep Medicine Rules for Assessing Sleep Depth in Clinical Practice”. en. In: *Journal of Clinical Sleep Medicine* 14.02 (Feb. 2018), pp. 205–213. ISSN: 1550-9389, 1550-9397. DOI: 10.5664/jcsm.6934. URL: http://jcsm.aasm.org/doi/10.5664/jcsm.6934 (visited on 02/17/2022).

[18] Thomas Penzel, Xiaozhe Zhang, and Ingo Fietze. “Inter-scorer Reliability between Sleep Centers Can Teach Us What to Improve in the Scoring Rules”. en. In: *Journal of Clinical Sleep Medicine* 09.01 (Jan. 2013), pp. 89–91. ISSN: 1550-9389, 1550-9397. DOI: 10.5664/jcsm.2352. URL: http://jcsm.aasm.org/doi/10.5664/jcsm.2352 (visited on 04/08/2022).

[19] Michael H. Bonnet et al. “The Scoring of Arousal in Sleep: Reliability, Validity, and Alternatives”. en. In: *Journal of Clinical Sleep Medicine* 03.02 (Mar. 2007), pp. 133–145. ISSN: 1550-9389, 1550-9397. DOI: 10.5664/jcsm.26815. URL: http://jcsm.aasm.org/doi/10.5664/jcsm.26815 (visited on 04/08/2022).

[20] Nancy A. Collop. “Scoring variability between polysomnography technologists in different sleep laboratories”. en. In: *Sleep Medicine* 3.1 (Jan. 2002), pp. 43–47. ISSN: 13899457. DOI: 10.1016/S1389-9457(01)00115-0. URL: https://linkinghub.elsevier.com/retrieve/pii/S1389945701001150 (visited on 04/25/2022).

[21] José S. Loredo et al. “Night-to-Night Arousal Variability and Interscorer Reliability of Arousal Measurements”. en. In: *Sleep* 22.7 (Oct. 1999), pp. 916–920. ISSN: 1550-9109, 0161-8105. DOI: 10.1093/sleep/22.7.916. URL: https://academic.oup.com/sleep/article-lookup/doi/10.1093/sleep/22.7.916 (visited on 04/25/2022).

[22] Donald Bliwise et al. “Measurement error in visually scored electrophysiological data: respiration during sleep”. en. In: *Journal of Neuroscience Methods* 12.1 (Nov. 1984), pp. 49–56. ISSN: 01650270. DOI: 10.1016/0165-0270(84)90047-5. URL: https://linkinghub.elsevier.com/retrieve/pii/0165027084900475 (visited on 04/25/2022).

[23] Susie Lord et al. “Interrater Reliability of Computer-Assisted Scoring of Breathing during Sleep”. en. In: *Sleep* 12.6 (Nov. 1989), pp. 550–558. ISSN: 0161-8105, 1550-9109. DOI: 10.1093/sleep/12.6.550. URL: https://academic.oup.com/sleep/article-lookup/doi/10.1093/sleep/12.6.550 (visited on 04/25/2022).

[24] Coralyn W. Whitney et al. “Reliability of Scoring Respiratory Disturbance Indices and Sleep Staging”. en. In: *Sleep* 21.7 (Oct. 1998), pp. 749–757. ISSN: 1550-9109, 0161-8105. DOI: 10.1093/sleep/21.7.749. URL: https://academic.oup.com/sleep/article-lookup/doi/10.1093/sleep/21.7.749 (visited on 04/25/2022).

[25] Michael J. Drinnan et al. “Interobserver Variability in Recognizing Arousal in Respiratory Sleep Disorders”. en. In: *American Journal of Respiratory and Critical Care Medicine* 158.2 (Aug. 1998), pp. 358–362. ISSN: 1073-449X, 1535-4970. DOI: 10.1164/ajrccm.158.2.9705035. URL: http://www.atsjournals.org/doi/abs/10.1164/ajrccm.158.2.9705035 (visited on 04/25/2022).

[26] Ulysses J. Magalang et al. “Agreement in the Scoring of Respiratory Events and Sleep Among International Sleep Centers”. en. In: *Sleep* 36.4 (Apr. 2013), pp. 591–596. ISSN: 0161-8105, 1550-9109. DOI: 10.5665/sleep.2552. URL: https://academic.oup.com/sleep/article/36/4/591/2595972 (visited on 04/25/2022).

[27] Samuel T. Kuna et al. “Agreement in Computer-Assisted Manual Scoring of Polysomnograms across Sleep Centers”. en. In: *Sleep* 36.4 (Apr. 2013), pp. 583–589. ISSN: 0161-8105, 1550-9109. DOI: 10.5665/sleep.2550. URL: https://academic.oup.com/sleep/article/36/4/583/2595970 (visited on 04/25/2022).

[28] Yun Ji Lee et al. “Interrater reliability of sleep stage scoring: a meta-analysis”. EN. In: *Journal of Clinical Sleep Medicine* (Jan. 2022). Publisher: American Academy of Sleep Medicine. DOI: 10.5664/jcsm.9538. URL: https://jcsm.aasm.org/doi/full/10.5664/jcsm.9538 (visited on 04/26/2022).

[29] Heidi Danker-Hopfe et al. “Interrater reliability for sleep scoring according to the Rechtschaffen & Kales and the new AASM standard”. en. In: *Journal of Sleep Research* 18.1 (2009). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2869.2008.00700.x, pp. 74–84. ISSN: 1365-2869. DOI: 10.1111/j.1365-2869.2008.00700.x. URL: http://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2869.2008.00700.x (visited on 04/26/2022).

[30] R. G. Norman et al. “Interobserver agreement among sleep scorers from different centers in a large dataset”. eng. In: *Sleep* 23.7 (Nov. 2000), pp. 901–908. ISSN: 0161-8105.

[31] Michael H. Silber et al. “The Visual Scoring of Sleep in Adults”. en. In: *Journal of Clinical Sleep Medicine* 03.02 (Mar. 2007), pp. 121–131. ISSN: 1550-9389, 1550-9397. DOI: 10.5664/jcsm.26814. URL: http://jcsm.aasm.org/doi/10.5664/jcsm.26814 (visited on 04/26/2022).

[32] Richard S Rosenberg. “the American Academy of sleep Medicine Inter-scorer Reliability Program: sleep stage scoring”. en. In: *Journal of Clinical Sleep Medicine* 9.1 (2013), p. 7.

[33] Thomas Penzel et al. “Digital Analysis and Technical Specifications”. en. In: *Journal of Clinical Sleep Medicine* 03.02 (Mar. 2007), pp. 109–120. ISSN: 1550-9389, 1550-9397. DOI: 10.5664/jcsm.26813. URL: http://jcsm.aasm.org/doi/10.5664/jcsm.26813 (visited on 04/25/2022).

[34] Reza Boostani, Foroozan Karimzadeh, and Mohammad Nami. “A comparative review on sleep stage classification methods in patients and healthy individuals”. en. In: *Computer Methods and Programs in Biomedicine* 140 (Mar. 2017), pp. 77–91. ISSN: 01692607. DOI: 10.1016/j.cmpb.2016.12.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169260716308276 (visited on 07/12/2022).

[35] Luay Fraiwan. “Automated sleep stage identification system based on time–frequency analysis of a single EEG channel and random forest classifier”. en. In: $computermethodsandprogramsinbiomedicine$ (), p. 10.

[36] Sheng-Fu Liang et al. “Automatic Stage Scoring of Single-Channel Sleep EEG by Using Multiscale Entropy and Autoregressive Models”. In: *IEEE Transactions on Instrumentation and Measurement* 61.6 (June 2012). Conference Name: IEEE Transactions on Instrumentation and Measurement, pp. 1649–1657. ISSN: 1557-9662. DOI: 10.1109/TIM.2012.2187242.

[37] Pierre Comon. “Independent component analysis, A new concept?” en. In: *Signal Processing*. Higher Order Statistics 36.3 (Apr. 1994), pp. 287–314. ISSN: 0165-1684. DOI: 10.1016/0165-1684(94)90029-9. URL: https://www.sciencedirect.com/science/article/pii/0165168494900299 (visited on 06/18/2023).

[38] Tarek Lajnef et al. “Learning machines and sleeping brains: Automatic sleep stage classification using decision-tree multi-class support vector machines”. en. In: *Journal of Neuroscience Methods* 250 (July 2015), pp. 94–105. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2015.01.022. URL: https://linkinghub.elsevier.com/retrieve/pii/S0165027015000230 (visited on 03/30/2023).

[39] Santosh Kumar Satapathy and D Loganathan. “Automated classification of multi-class sleep stages classification using polysomnography signals: a nine- layer 1D-convolution neural network approach”. en. In: *Multimedia Tools and Applications* 82.6 (Mar. 2023), pp. 8049–8091. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-022-13195-2. URL: https://link.springer.com/10.1007/s11042-022-13195-2 (visited on 02/26/2023).

[40] Ashish Vaswani et al. “Attention is All You Need”. In: 2017. URL: https://arxiv.org/pdf/1706.03762.pdf (visited on 11/19/2022).

[41] Emadeldeen Eldele et al. “An Attention-Based Deep Learning Approach for Sleep Stage Classification With Single-Channel EEG”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021). Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering, pp. 809–818. ISSN: 1558-0210. DOI: 10.1109/TNSRE.2021.3076234.

[42] Akara Supratak et al. “DeepSleepNet: A Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.11 (Nov. 2017). Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering, pp. 1998–2008. ISSN: 1558-0210. DOI: 10.1109/TNSRE.2017.2721116.

[43] Siddharth Biswal et al. “Expert-level sleep scoring with deep neural networks”. en. In: *Journal of the American Medical Informatics Association* 25.12 (Dec. 2018), pp. 1643–1650. ISSN: 1067-5027, 1527-974X. DOI: 10.1093/jamia/ocy131. URL: https://academic.oup.com/jamia/article/25/12/1643/5185596 (visited on 04/20/2023).

[44] Antoine Guillot et al. “Dreem Open Datasets: Multi-Scored Sleep Datasets to Compare Human and Automated Sleep Staging”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28.9 (Sept. 2020). Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering, pp. 1955–1965. ISSN: 1558-0210. DOI: 10.1109/TNSRE.2020.3011181.

[45] Yannick Roy et al. “Deep learning-based electroencephalography analysis: a systematic review”. en. In: *Journal of Neural Engineering* 16.5 (Oct. 2019), p. 051001. ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2552/ab260c. URL: https://iopscience.iop.org/article/10.1088/1741-2552/ab260c (visited on 04/28/2022).

[46] Mathias Perslev et al. “U-Sleep: resilient high-frequency sleep staging”. en. In: *npj Digital Medicine* 4.1 (Dec. 2021), p. 72. ISSN: 2398-6352. DOI: 10.1038/s41746-021-00440-5. URL: http://www.nature.com/articles/s41746-021-00440-5 (visited on 05/20/2022).

[47] Mathias Perslev. *U-Time & U-Sleep.* original-date: 2019-01-09T12:18:58Z. Apr. 2022. URL: https://github.com/perslev/U-Time (visited on 04/29/2022).

[48] Orestis Tsinalis, Paul M. Matthews, and Yike Guo. “Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders”. en. In: *Annals of Biomedical Engineering* 44.5 (May 2016), pp. 1587–1597. ISSN: 0090-6964, 1573-9686. DOI: 10.1007/s10439-015-1444-y. URL: http://link.springer.com/10.1007/s10439-015-1444-y (visited on 04/28/2022).

[49] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. ISSN: 1533-7928. URL: http://jmlr.org/papers/v12/pedregosa11a.html (visited on 05/31/2023).

[50] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library.* Pages: 8024–8035 Publication Title: Advances in Neural Information Processing Systems 32 original-date: 2016-08-13T05:26:41Z. 2019. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf (visited on 07/25/2023).

[51] conda contributors. *conda: A system-level, binary package and environment manager running on all major operating systems and platforms.* original-date: 2012-10-15T22:08:03Z. July 2023. URL: https://github.com/conda/conda (visited on 07/25/2023).

[52] Eric Larson et al. *MNE-Python.* Feb. 2023. DOI: 10.5281/zenodo.7671973. URL: https://zenodo.org/record/7671973 (visited on 05/25/2023).

[53] Alexandre Gramfort et al. “MNE software for processing MEG and EEG data”. en. In: *NeuroImage* 86 (Feb. 2014), pp. 446–460. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2013.10.027. URL: https://linkinghub.elsevier.com/retrieve/pii/S1053811913010501 (visited on 05/25/2023).

[54] Khald Ali I. Aboalayon et al. “Sleep Stage Classification Using EEG Signal Analysis: A Comprehensive Survey and New Investigation”. en. In: *Entropy* 18.9 (Sept. 2016). Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 272. ISSN: 1099-4300. DOI: 10.3390/e18090272. URL: https://www.mdpi.com/1099-4300/18/9/272 (visited on 08/11/2022).

[55] Panteleimon Chriskos et al. “A review on current trends in automatic sleep staging through bio-signal recordings and future challenges”. en. In: *Sleep Medicine Reviews* 55 (Feb. 2021), p. 101377. ISSN: 10870792. DOI: 10.1016/j.smrv.2020.101377. URL: https://linkinghub.elsevier.com/retrieve/pii/S1087079220301209 (visited on 04/28/2022).

[56] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. en. In: *Nature Methods* 17.3 (Mar. 2020), pp. 261–272. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0686-2. URL: https://www.nature.com/articles/s41592-019-0686-2 (visited on 05/31/2023).

[57] F Perrin et al. “Mapping of scalp potentials by surface spline interpolation”. en. In: *Electroencephalography and Clinical Neurophysiology* 66.1 (Jan. 1987), pp. 75–81. ISSN: 00134694. DOI: 10.1016/0013-4694(87)90141-6. URL: https://linkinghub.elsevier.com/retrieve/pii/0013469487901416 (visited on 05/20/2023).

[58] Mats Svantesson et al. “Virtual EEG-electrodes: Convolutional neural networks as a method for upsampling or restoring channels”. en. In: *Journal of Neuroscience Methods* 355 (May 2021), p. 109126. ISSN: 0165-0270. DOI: 10.1016/j.jneumeth.2021.109126. URL: https://www.sciencedirect.com/science/article/pii/S0165027021000613 (visited on 05/20/2023).

[59] M Hamed Mozaffari and Li-Lin Tay. “A Review of 1D Convolutional Neural Networks toward Unknown Substance Identification in Portable Raman Spectrometer”. en. In: ().

[60] Lucas C. Parra et al. “Recipes for the linear analysis of EEG”. en. In: *NeuroImage* 28.2 (Nov. 2005), pp. 326–341. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2005.05.032. URL: https://linkinghub.elsevier.com/retrieve/pii/S1053811905003381 (visited on 07/26/2023).

[61] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. en. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, Aug. 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: https://dl.acm.org/doi/10.1145/2939672.2939785 (visited on 07/26/2023).

[62] Margherita Grandini, Enrico Bagli, and Giorgio Visani. *Metrics for Multi-Class Classification: an Overview*. en. arXiv:2008.05756 [cs, stat]. Aug. 2020. URL: http://arxiv.org/abs/2008.05756 (visited on 05/22/2023).

[63] Aniruddha Bhandari. *Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)*. en. Apr. 2020. URL: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/ (visited on 05/22/2023).

[64] Frank Krüger. “Activity, Context, and Plan Recognition with Computational Causal Behaviour Models”. PhD thesis. Dec. 2016.

[65] Hyungjik Kim, Seung Min Lee, and Sunwoong Choi. “Automatic sleep stages classification using multi-level fusion”. en. In: *Biomedical Engineering Letters* 12.4 (Nov. 2022), pp. 413–420. ISSN: 2093-985X. DOI: 10.1007/s13534-022-00244-w. URL: https://doi.org/10.1007/s13534-022-00244-w (visited on 04/21/2023).

[66] Christian O’Reilly et al. “Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research”. en. In: *Journal of Sleep Research* 23.6 (2014). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jsr.12169, pp. 628–635. ISSN: 1365-2869. DOI: 10.1111/jsr.12169. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/jsr.12169 (visited on 10/27/2022).

[67] Yuhei Ichimaru and George B Moody. *MIT-BIH Polysomnographic Database*. 1992. DOI: 10.13026/C23K5S. URL: https://physionet.org/content/slpdb/ (visited on 04/30/2022).

[68] Mario Giovanni Terzano et al. *CAP Sleep Database*. 2001. DOI: 10.13026/C2VC79. URL: https://physionet.org/content/capslpdb/ (visited on 04/30/2022).

[69] Diego Alvarez-Estevez and Roselyne Rijsman. *Haaglanden Medisch Centrum sleep staging database*. DOI: 10.13026/T79Q-FR32. URL: https://physionet.org/content/hmc-sleep-staging/1.1/ (visited on 04/30/2022).

[70] Walter McNicholas et al. *St. Vincent's University Hospital / University College Dublin Sleep Apnea Database*. en. 2004. DOI: 10.13026/C26C7D. URL: https://physionet.org/content/ucddb/ (visited on 04/29/2022).

[71] Bastiaan Kemp et al. *The Sleep-EDF Database [Expanded]*. 2018. DOI: 10.13026/C2X676. URL: https://physionet.org/content/sleep-edfx/ (visited on 04/30/2022).

[72] Stephanie Devuyst. *The DREAMS Databases and Assessment Algorithm*. eng. Jan. 2005. DOI: 10.5281/zenodo.2650142. URL: https://zenodo.org/record/2650142 (visited on 07/03/2022).

[73] G. Klosh et al. "The SIESTA project polygraphic and clinical database". In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (May 2001), pp. 51–57. ISSN: 1937-4186. DOI: 10.1109/51.932725.

[74] P. Rappelsberger et al. "Das Projekt SIESTA". de. In: *Klinische Neurophysiologie* 32.02 (June 2001), pp. 76–88. ISSN: 1434-0275, 1439-4081. DOI: 10.1055/s-2001-16206. URL: http://www.thieme-connect.de/DOI/DOI?10.1055/s-2001-16206 (visited on 07/01/2022).

[75] Guo-Qiang Zhang et al. "The National Sleep Research Resource: towards a sleep data commons". eng. In: *Journal of the American Medical Informatics Association: JAMIA* 25.10 (Oct. 2018), pp. 1351–1358. ISSN: 1527-974X. DOI: 10.1093/jamia/ocy064.

[76] S. F. Quan et al. "The Sleep Heart Health Study: design, rationale, and methods". eng. In: *Sleep* 20.12 (Dec. 1997), pp. 1077–1085. ISSN: 0161-8105.

[77] Sirvan Khalighi et al. "ISRUC-Sleep: A comprehensive public dataset for sleep researchers". In: *Computer Methods and Programs in Biomedicine* 124 (Nov. 2015). DOI: 10.1016/j.cmpb.2015.10.013.

[78] Ximena Omlin et al. "The Effect of a Slowly Rocking Bed on Sleep". en. In: *Scientific Reports* 8.1 (Dec. 2018), p. 2156. ISSN: 2045-2322. DOI: 10.1038/s41598-018-19880-3. URL: http://www.nature.com/articles/s41598-018-19880-3 (visited on 07/10/2022).