Automated Sleep Scoring, Deep Learning and Physician Supervision

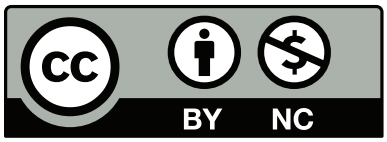
Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Luigi FIORILLO

von Italien

Leiter der Arbeit:
Prof. Dr. Paolo FAVARO
Institut für Informatik
Prof. Dr. Francesca Dalia FARACI
Institute of Digital Technologies for Personalized Healthcare SUPSI



Automated Sleep Scoring, Deep Learning and Physician Supervision

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Luigi FIORILLO

von Italien

Leiter der Arbeit:
Prof. Dr. Paolo FAVARO
Institut für Informatik
Prof. Dr. Francesca Dalia FARACI
Institute of Digital Technologies for Personalized Healthcare SUPSI

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Bern, 28.10.2022

Der Dekan
Prof. Dr. Marco Herwegh

Abstract

Automated Sleep Scoring,
Deep Learning and Physician Supervision

Luigi FIORILLO, Ph.D. in Computer Science

Universität Bern, 2021

Sleep plays a crucial role in human well-being. Polysomnography is used in sleep medicine as a diagnostic tool, so as to objectively analyze the quality of sleep. Sleep scoring is the procedure of extracting sleep cycle information from the whole-night electrophysiological signals. The scoring is done worldwide by the sleep physicians according to the official American Academy of Sleep Medicine (AASM) scoring manual. In the last decades, a wide variety of deep learning based algorithms have been proposed to automatise the sleep scoring task. In this thesis we study the reasons why these algorithms fail to be introduced in the daily clinical routine, with the perspective of bridging the existing gap between the automatic sleep scoring models and the sleep physicians. In this light, the primary step is the design of a simplified sleep scoring architecture, also providing an estimate of the model uncertainty. Beside achieving results on par with most up-to-date scoring systems, we demonstrate the efficiency of ensemble learning based algorithms, together with label smoothing techniques, in both enhancing the performance and calibrating the simplified scoring model. We introduced an uncertainty estimate procedure, so as to identify the most challenging sleep stage predictions, and to quantify the disagreement between the predictions given by the model and the annotation given by the physicians. In this thesis we also propose a novel method to integrate the inter-scorer variability into the training procedure of a sleep scoring model. We clearly show that a deep learning model is able to encode this variability, so as to better adapt to the consensus of a group of scorers-physicians. We finally address the generalization ability of a deep learning based sleep scoring system, further studying its resilience to the sleep complexity and to the AASM scoring rules. We can state that there is no need to train the algorithm strictly following the AASM guidelines. Most importantly, using data from multiple data centers results in a better performing model compared with training on a single data cohort. The variability among different scorers and data centers needs to be taken into account, more than the variability among sleep disorders.

Acknowledgements

Foremost, I would like to thank both my Ph.D. advisors Francesca Faraci, head of the BSP group in Lugano, and Paolo Favaro, head of the CVG team in Bern.
I am highly grateful for all the opportunities and the outstanding technical and emotional support Francesca gave me during this four years journey. She challenged me to always push further. She constantly motivated me along the way, she gave me all the freedoms with no pressure and she gave me as much time as I needed to accomplish my goals. I am extremely thankful for Paolo valuable advice, the inspiring discussions and his effort in getting me to discover and to study in depth the deep learning world. Among other things, he taught me, or at least I hope I learned, the art of *summarize it all in a nutshell*.

I want to express my thanks to Filippo Molinari and Torsten Braun for their availability as examiners. I am very glad Filippo agreed to serve as an examiner, I greatly admired his works and I really appreciated his lectures at the Polytechnic of Turin during my master's degree.

A huge thank you goes out to all the members and researchers of the MediTech Institute and of the BSP group I had the pleasure to work with: Alessandro Puiatti, Giuliana Monachino, Stefano Scafa, Davide Pedroncelli, Beatrice Zanchi, Radoslava Svihrova, Michal Bechny, Luis Germain Arango. I really enjoyed the work atmosphere in the group, all the coffee/lunch breaks and the fun time we had outside the working environment. A special thank you goes to Alessandro, he was my MSc thesis mentor and he pushed me from the very beginning to pursue a Ph.D. A big thank you goes to Giuliana and Davide, without their great help I would not have been able to finish the last two papers as quickly as we did. A huge thank you also goes to Michela Papandrea, Alberto Vancheri, Luca Luceri, Felipe Cardoso and Michael Wand, for all the interesting discussions we had and the good time we spent.

I would like to thank all the members of the CVG lab in Bern, which I regrettably did not get to know as well, but who taught me and inspired me so much during our weekly CVG lab meetings.

Thanks also to the members of the Sleep Wake Epilepsy Center | NeuroTec at the Inselspital, Bern University Hospital, I had the pleasure to work with: Corinne Roth, Julia van der Meer, Markus Schmidt and Claudio Bassetti, for their valuable clinical support and without whom I would never have been able to perform the analyses on such a large amount of data.

A huge thanks goes to Martin Monti, head of the MontiLab at UCLA in Los Angeles, and to all his group with whom I collaborated during my Ph.D. mobility-visiting program, and with whom I spent one of the most inspiring and exciting times of the last four years.

My mother, my father, my little brother, my two sisters and all my close friends have definitely been the backbone of the entire journey. They are and they will be the ones who will continue to support me throughout my career and my life. Without them, no achievement would have the same weight.

Last but not least, I would like to thank my soon to be wife Morena. She pushed me beyond my limits and she taught me to always chase my dreams.

Contents

List of Figures

List of Tables

Alle mie nonne...

Chapter 1

Introduction

Sleep disorders represent a significant and increasing public health problem. A considerable proportion of the world population is suffering from serious sleep disorders and requires medical attention [1]. Since its origin, in the late 1950s, polysomnography (PSG) has been at the centre of several investigations, aiming at simplifying the related scoring procedure so as to better analyze the quality of sleep and the common sleep pathologies, *e.g.*, sleep breathing disorders, narcolepsy, sleep-related movement disorders. A PSG is the whole night recording of several biosignals related to sleep. Brain activity (electroencephalogram), eye movements (electrooculogram), muscle activity or skeletal muscle activation (electromiogram derivations for chin and legs), body position (video camera and accelerometer), heart rhythm (electrocardiogram), breathing functions (respiratory airflow, oxygen saturation, respiratory effort indicators) and other vital parameters are monitored overnight. A PSG typically requires that the patients sleep overnight at the hospital while their bio-physiological signals are recorded.

The sleep scoring is the procedure of extracting information from the PSG signals. Sleep stages, arousals, respiratory events, movements and cardiac events have to be correctly identified. Wakefulness and sleep phases (*i.e.*, stages 1, 2, 3 and rapid eye movement) can be mainly described by the following bio-signals: electroencephalography (EEG), electrooculography (EOG) and electromyography (EMG). Clinical sleep scoring involves a visual review of overnight polysomnograms by a human expert, that may require up to two hours of tedious repetitive work. This can explain why the search for simplifying and speeding up the sleep scoring work begun already in the late 1960s.

The scoring is done worldwide accordingly to official standards, *i.e.*, the American Academy of Sleep Medicine (AASM) scoring manual for adults, children and infants [2]. It could appear then a suitable task for modern artificial intelligence (AI) algorithms. Many different techniques and approaches have been proposed and tested, reaching very good results in terms of overall accuracy. Machine learning (ML) algorithms have been applied to sleep scoring for many years. As a result, several software products offer nowadays automated or semi-automated scoring services. Very recently, thanks to the increased computational power, deep learning (DL) based algorithms have also been employed with promising results. ML/DL algorithms can undoubtedly reach a high accuracy in specific situations, but there are many difficulties in their introduction in the daily clinical routine. The vast majority of the sleep physicians do not use them. The high inter-scorer variability (agreement of about 70-80%), and the low intra-scorer agreement (about 90%) [3]–[5], may partially explain the low acceptance of the automated scoring systems.

1.1 Motivation and Challenges

The automatic sleep scoring on healthy people, on both adults and children, has been practically solved by using EEG, EOG and EMG biosignals. Indeed, different scoring algorithms results in a higher consensus level compared to the averaged inter-scorer agreement. Nevertheless, even in simple applications (*e.g.*, on a distribution of healthy subjects), these algorithms fail to be exploited in the daily routine. Clearly, the real needs of the single physician are still not completely understood.

Below we report the two main reasons why physicians do not use the ML/DL scoring algorithms, setting the background for the analysis we present in this thesis. Our final perspective is to bridge the existing gap between the automatic sleep scoring models and the sleep physicians.

Scorer personalization. In our view, the major challenge lies in customizing the sleep scoring systems for each physician. The common practice is to train a scoring system with datasets where each whole night recording is annotated by a single physician or sleep lab. Only in a few cases, we have datasets where each recording is annotated by multiple physicians and/or from different sleep labs. Thus, typically, the model emulate the scoring procedure of a single physician/lab or of the consensus of a group of physicians. Most likely a new physician (*i.e.*, the end-user) will disagree with the single scorer or with the group of scorers. Hence, the need to build a closed-loop interaction between the scoring algorithm and the end-user. The prospective is to personalise a sleep scoring algorithm on the end-user "taste", eventually fine-tuning the model to the scoring rules of the new scorer. However, tackling such a challenge requires first to correctly identify the disagreement between the predictions given by the model and the end-user annotations. Hence, the first need and our primary challenge is to be able to quantify the uncertainty of a sleep scoring algorithm. The predictions with low confidence (lowest probability values in output) are more likely to match with the disagreed ones. However, unfortunately, the wrong prediction are often associated with high probability values. Methods quantifying wrong decisions needs to be implemented.

Data heterogeneity. A big challenge involves both the mismatch of sleep recordings coming from different sources or data domains, and the resulting heterogeneity among these recordings. This leads to different hardware or subjects with different demographic and sleep disorders, even in the same data cohort. In a real-case scenario, the performance of a scoring algorithm on a PSG coming from an unseen data distribution (*e.g.*, different data domain) would drastically decrease. Changes in numbers and positions of the available channels, or changes in type of sleep disorders - worst case scenario neurological disorders - may require different algorithmic solutions. It raises the need to first explore and then adapt existing scoring architectures on new data domain via transfer learning or domain adaptation techniques. Even in the same data cohort, it would be extremely interesting to investigate the impact of different data distributions (*e.g.*, changes in well-known patient demographic such as the chronological age) on the performance of the scoring algorithms.

1.2 Contributions

In this thesis we study the reasons why the existing sleep scoring algorithms fail to be introduced in the daily clinical routine, with the vision of bridging the gap between the automatic scoring models and the sleep physicians.

With the ultimate goal of assessing the disagreement between the predictions given by the DL based sleep scoring algorithm and the end-user annotations, we first propose to simplify an existing state-of-the-art scoring architecture while maintaining its high performance. We exploit ensemble learning based algorithms together with label smoothing techniques, as to enhance the calibration of the model and to further enhance the performance of the scoring architecture. Then we introduce an uncertainty estimate procedure to identify the most challenging sleep stage predictions, so as to quantify the number of predictions in disagreement with the physicians, *i.e.*, to quantify the misclassified sleep stages. We demonstrate the efficiency of these methodologies on different scoring architectures and sleep databases. In a follow-up study we also validate the hypothesis that our simplified sleep scoring architecture is actually able to learn from a multi-scored dataset. The scoring algorithm is able to learn the inter-scorer variability, so as to adapt to the scoring consensus distribution of multiple physicians.

In order to study the generalization ability of a sleep scoring algorithm, we exploit a powerful and recently proposed U-Net inspired architecture, testing it on different large-scale-heterogeneous data cohorts, *i.e.*, 28528 polysomnography studies from 13 different clinical sleep labs. We highlight two very interesting properties of a DL based scoring algorithm: there is no need to train it following the strict AASM scoring guidelines, and there is no need to train it with additional chronological age-related information. We finally demonstrate that the variability among different sleep data centers (*e.g.*, hardware, scoring rules etc.) needs to be taken into account, more than the variability among different subjects with different sleep disorders.

1.2.1 Chapter Outline

Chapter 2: Background. We provide a general overview about the clinical sleep scoring, and the ML algorithms proposed to automatize the scoring procedure. In Chapter 2 we focus on the very latest approaches that exploit DL based architectures to tackle the clinical task. We further discuss about the existing barriers to the introduction of the automated scoring in the clinical practice.

Chapter 3: Automated sleep scoring, temporal dependency and recurrent neural networks. Most of the DL based sleep scoring architectures exploit recurrent neural networks (RNNs) to model the temporal dynamic behaviour of sleep. In Chapter 3, starting from a well-known state-of-the-art architecture, we propose to replace the recurrent layers with simple fully connected layers. The results suggest that a simple feed forward architecture achieves comparable performance to those using the RNNs. Thus, the reason why the architectures succeed in encoding the temporal dynamic behaviour (*e.g.*, stage transitions) may not necessarily relate to the recurrent blocks itself, but to the temporal context we give as input.

Chapter 4: A simplified sleep scoring model with uncertainty estimates. Existing DL based sleep scoring algorithms exploit computationally demanding architectures

and process lengthy time sequences in input. Only few of these architectures provide an estimate of the model uncertainty. In Chapter 4 we propose DeepSleepNet-Lite, a simplified and lightweight scoring architecture, processing only 90-second EEG sequences in input. We exploit, for the first time in sleep scoring, the Monte Carlo dropout along with the label smoothing technique. We show the efficiency of these techniques in calibrating and enhancing the performance of our model. We also demonstrate the efficiency of the proposed uncertainty estimate procedure: it is able to identify the most challenging sleep stage predictions.

Chapter 5: Exploitation of the multi-scored databases in automated sleep scoring. The visual scoring of a PSG is a highly subjective procedure. Most of the existing DL based sleep scoring algorithms are trained using datasets where each recording is annotated by a single scorer. Even when annotations from two or more scorers are available, the architectures are trained on the single one-hot encoded label (*i.e.*, scorer consensus). The averaged scorer's subjectivity is transferred into the model, losing information about the internal variability among different scorers. In Chapter 5 we propose to use label smoothing along with the soft-consensus distribution to consider the multiple-labels, *i.e.*, the annotation from different physicians, into the training procedure of the model. The results suggest that our approach enables the model to better adapt to the consensus of the group of scorers.

Chapter 6: U-Sleep: resilient to AASM guidelines. AASM guidelines are the results of decades of efforts to try to standardize the sleep scoring procedure as to have a commonly used methodology. The guidelines cover several aspects from the technical specifications to the sleep scoring rules. Any DL based sleep scoring algorithm is trained on recordings annotated by sleep physicians according to the AASM manuals. Clinical knowledge and guidelines have been exploited to support the algorithms in solving the scoring task. In Chapter 6 we show that a DL based sleep scoring algorithm may not need to fully exploit the clinical knowledge or to strictly follow the AASM guidelines. Specifically, we demonstrate that U-Sleep, a state-of-the-art sleep scoring algorithm, can be strong enough to solve the scoring task even by using clinically non-recommended or non-conventional derivations, and with no need to exploit information about the chronological age of the subjects. We finally strengthen a well-known finding that using data from multiple data centers always results in a better performing model compared with training on a single cohort. We show that this latter statement is still valid even by increasing the size and the heterogeneity of the single data cohort.

Each chapter is associated with one of the papers as in the following list:

- Chapter 2 - "Automated sleep scoring: A review of the latest approaches" [6], published in Sleep Medicine Reviews, 2019.
- Chapter 3 - "Temporal dependency in automatic sleep scoring via deep learning based architectures: An empirical study" [7], published in EMBC, 2020.
- Chapter 4 - "DeepSleepNet-Lite: A Simplified Automatic Sleep Stage Scoring Model with Uncertainty Estimates" [8], published in IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2021.
- Chapter 5 - "Multi-Scored Sleep Databases: How to Exploit the Multiple-Labels in Automated Sleep Scoring", in Sleep, 2022. *(submitted)*

- Chapter 6 - "U-Sleep: resilient to AASM guidelines", in Nature Portfolio Journal digital medicine, 2022. *(submitted)*

Chapter 2

Background

2.1 Visual Sleep Scoring Procedure

The polysomnographic record of sleep is usually divided into 30-second epochs, starting from the lights-off event. This time interval is a heritage from old PSG machines where a paper speed of 10mm/s gave an output page of a 30-second timespan. During a visual analysis each epoch is assigned a stage, and if two or more stages coexist during a single epoch the stage comprising the majority of the 30 seconds is scored (an example in Figure 2.1).

In 1968 the first manual to standardize terminology and rules of this procedure was published by Rechtschaffen and Kales (R&K) [9]. It categorized sleep into seven distinct stages: wakefulness, stages 1, 2, 3, 4, rapid eye movement (REM) sleep and movement time (MT) stage. These rules were adopted worldwide until 2007, when the AASM updated the scoring manual [2]. The AASM standard manual for the scoring of sleep and associated events is designed to cover all aspects of the PSG, from the technical ones (parameters, assessment protocols, filtering, etc.), to its execution, the analytic scoring (sleep staging, arousals, cardiac, movement, and respiratory signals), and the final interpretation of PSG results. The number of stages was reduced to five: wakefulness W, stage N1, stage N2, stage N3 (formerly 3 and 4 sleep stages), and stage R (formerly REM sleep stage). MT stage was abolished, and it was decided to score an epoch with a major body movement as wake if any part of the epoch shows alpha rhythm, or if a wake epoch either precedes or follows the epoch in question. Otherwise, the epoch is scored as the same stage as the epoch that follows it. Almost every year there is a new version of the AASM manual with usually a few updates. Recommended EEG derivations are F4-M1, C4-M1, O2-M1, while other accepted derivations are Fz-Cz, Cz-Oz, C4-M1. EEG can be contaminated by other electrophysiological signals, as for example ECG, EOG, EMG and pulse-oxymetry signal. Movement artifacts are also often present and need to be addressed. EEG is conventionally described in terms of its frequency components. The main ones are delta (0.5-4Hz), theta (4-8Hz), alpha (8-12Hz), and beta (12-35Hz). Waves in the frequency range 0.5-2Hz and peak-to-peak amplitude >75μV are considered slow wave activity. Sleep spindles (train of distinct waves in the 12-14Hz range, lasting for more than 0.5-seconds), K-complexes (sharp negative waves followed by a smooth, positive waves longer than 0.5-seconds) and vertex sharp waves (negative-going bursts of less than 0.5-seconds) are also introduced to better describe the EEG. Scoring rules are based on the recognition of EEG frequencies and on the presence of certain pattern, but applying these rules can lead to unexpected complexity, especially in unhealthy subjects.

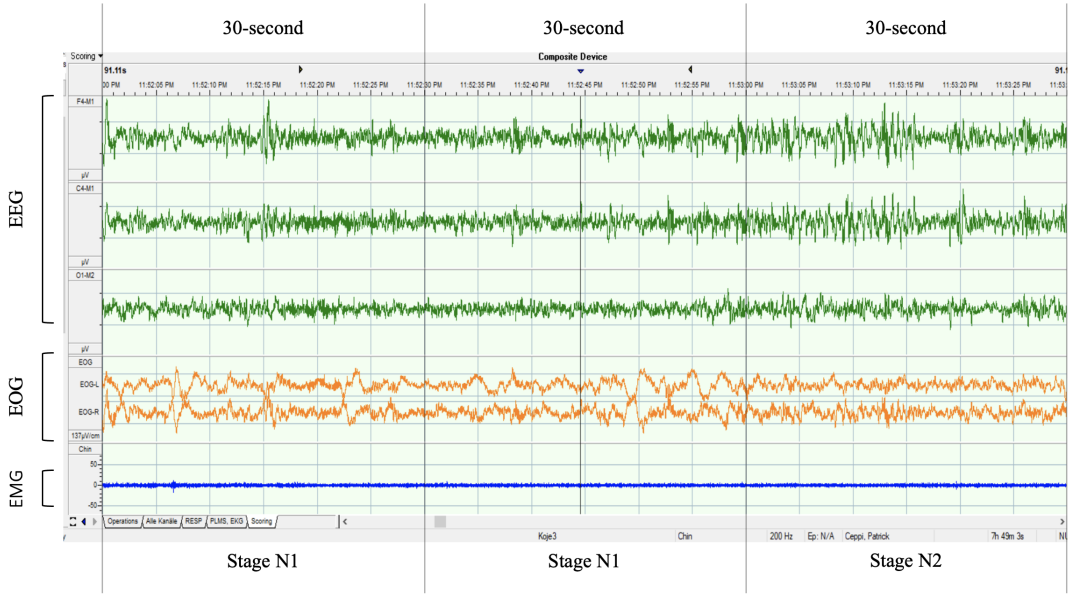


FIGURE 2.1: **An example of polysomnographic sleep epochs.** An example of a polysomnographic record of sleep divided into 30-second epochs; each epoch is assigned a sleep stage.

Sleep stages progress cyclically from N1 through R, then begin again with stage N1. A full sleep cycle takes on average from 90 to 110 minutes, with each stage lasting between 5 to 15 minutes. The first sleep cycles present relatively short REM sleeps and long periods of deep sleep. The characteristics of the sleep phases and the scoring rules accordingly to the AASM are summarized in the following list.

- ***Stage W***: it is characterized by the presence of alpha rhythm in the EEG signal, usually over the occipital region, and/or any of the following events: eye blinking, rapid eye movements with normal or high chin muscle activity (with signal frequency higher than 30Hz), reading eye movements.

- ***Stage N1***: it shows slow eye movements and it can be easily disrupted leading to awakenings or arousals. EEG signal amplitude does not exceed 200mV with frequencies within 2-7Hz. Alpha components should not exceed 50% of the total spectral band, and vertex sharp waves are often seen during transitions from other stages to N1. Slow eye movements can be visible in the EOG, and EMG level should be lower than in the previous stage. N1 continues until there is evidence of another stage. N1 usually covers 5% of the sleep time.

- ***Stage N2***: awakenings or arousals are not so common as in N1 and the slow-moving eye starts to disappear. Sleep spindles and K complexes may appear. N2 should be scored if during the last half of the previous epoch or the first half of the actual one there are either one or more K-complex or one or more trains of spindle, and it should continue to be scored N2 (also without spindles and K-complexes), until a new stage appear. N2 normally covers 50% of the sleep time.

- ***Stage N3***: it is the deep restorative sleep. Delta waves and slow waves are predominant in the EEG signal. Awakenings or arousals are rare. Spindles

and K-complex may appear. N3 should be scored if more than 20% of the epoch consists of slow waves. N3 covers usually 20% of the sleep time.

- ***Stage REM***: the dreaming stage. Eye movements are rapid and brain waves are more active than in N2 and in N3. Awakenings and arousals can occur more easily in REM. The EEG has low voltage, mixed frequency and possible sawtooth waves, EMG is at its lowest level, episodic REMs usually lasting less than 500ms appears in the EOG. A stage should continue to be scored as REM until one or more of the following occur: a transition to stage W or N3 appears; chin EMG muscle tone increases; a K-complex without arousal or a spindle occurs in the first half of the epoch with no REMs. This stage normally covers 20-25% of the sleep time.

The sleep staging procedure may be quite complex: many parameters have to be considered at the same time; previous and future epoch scoring has to be taken into account as well. The heterogeneity of the subjects and of the sleep epochs may be quite difficult to be comprehensively described in a manual, which generates uncertainty in the scoring procedure and leads to different interpretations of the same signal from different scorers. Subjects with specific sleep disorders can be more challenging to be scored than healthy ones. It has also to be highlighted that some errors may be more costly than others. N2 is very often considered as a transition phase between light and deep sleep, consequently if N2 percentage is a little higher or a little lower, the impact on the "big picture" of the sleep analysis will be minimal. The presence of sleep apneas, parasomnias, periodic limb movement and of other sleep abnormalities will still be examined if N2 is confused with N1 or N3. Instead, if the error is related to wakefulness all the scoring process will be impacted, as the presence of sleep abnormalities will not be considered. Inter-rater variability studies show how agreement can vary among stages [10]. Rosemberg et al. [3] compared a large number of scorers (>2500): the agreement was higher than 80% for REM, N2 and W, but it dropped for N3 (67%) and N1 (63%), the overall agreement was of about 83%. Human scorers' discrepancies occur mainly in the judgment of transitions between two different stages. This is not surprisingly as AASM rules are trying to characterize a continuum physiological process with fixed stages.

2.2 Automated Sleep Scoring by Artificial Intelligence

AI consists of the emulation of human intelligence processes performed by machines. ML is an application of AI that provides systems the ability to automatically learn and improve from experience.
ML algorithms can then replicate human intelligence processes to assist and simplify manual process. This powerful approach should then be suitable for sleep scoring, which could be considered as a tedious, repetitive classification work based on the observation of standardized rules. Two main approaches of AI might potentially address the automatic sleep stage classification (ASSC) problem: learning processes based on features extracted starting from the knowledge of the experts (*i.e.*, shallow learning), and learning processes that start directly from the raw data (*i.e.*, DL).

2.2.1 Shallow Learning Approach

In a ML workflow, the main steps are data pre-processing, feature extraction, feature selection/dimensionality reduction and classification. The pre-processing phase allows the detection of bias, noise or artifact present in the PSG raw signals. The

features extraction, feature selection and dimensionality reduction steps allow to identify the most relevant information. The last classification phase uses all the information for sleep stage identification.

Feature extraction and feature selection techniques

The feature extraction procedure starts from the measured data and derives values (*i.e.*, features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to a better human interpretation. A feature is an individual measurable property or characteristic of the PSG; an example can be the time-domain signal power over the entire epoch. Feature extraction techniques can be linear and non-linear and can be grouped into three major categories: temporal domain methods, frequency domain methods and hybrid of temporal and frequency domain methods [11], [12]. Among the most recent works, the standard statistics in time domain, the non-parametric analysis in frequency domain and the wavelet transform in time-frequency domain are the most used techniques for ASSC [13].
In some cases the extracted features are redundant or generate a dataset with a high dimensionality. Dimensionality reduction is a process that can reduce the number of features, focusing on the most significant ones. The most widely used approaches are the principal component analysis, for dimensionality reduction, and the sequential floating search method for feature selection [13]. These techniques allow representing data in a reduced dimensional space maintaining almost the same information, resulting in increased performance of the classifier [14].

Machine learning classifiers

The ML classification techniques used in automatic sleep stage identification - shallow learning approach - are manifold. Several reviews have exhaustively analyzed feature-based approaches. In particular, Ronzhina et al. [15] reviewed classification systems using artificial neural networks (ANN) in automatic sleep scoring. The reported ANN based scoring system performance varies within a broad range of accuracy, depending on the recognized stages. Şen et al. [12] carried out a comparative study trying to identify the most effective features and the most efficient algorithm to classify the sleep stages. They propose a methodology that can reach an overall accuracy of 98%. Radha et al. [16] also tried to identify optimal ML and signal processing methods, focusing on online sleep staging and a single EEG channel. They concluded that spectral linear features, epoch duration between 18 and 30 seconds, and a random forest classifier lead to optimal classification performance while ensuring real-time online operation. In the comprehensive survey of Aboalayon et al. [13] several sleep stage classification techniques using EEG signals have been reported and compared, with accuracy ranging from 70 to 94%. They have also presented their own approach based on novel features and using 10-second epochs claiming to reach an average accuracy of 93%.
It is important to note that comparing the performance of different approaches is a quite complex task. Sleep stages considered, extracted features, datasets and channels, classification algorithms, validation methods adopted and evaluation metrics reported have to be taken into consideration. For a better comparison, some researchers have reapplied classification approaches to the same dataset. For example, in a recent work, Boostani et al. [17] carried out a comparative review of several ML classification techniques used in ASSC. They selected five classification

approaches [18]–[22] and reapplied them on public datasets containing PSG data of healthy and unhealthy subjects. They tested various combinations of extracted features and classification techniques in order to find the best one in predicting sleep stages correctly. The random forest classifier together with the entropy of wavelet coefficients proved to be the best combination, reporting percentages of accuracy of 87% in healthy subjects and 69% in patients.

DL based algorithms can also be applied in the feature-based workflow with good results [23]–[25], but they exert their full potential when applied directly to raw data, as presented in the next paragraph.

2.2.2 Deep Learning Approach

DL is part of a broader family of ML methods; it is based on learning data representations, as opposed to task-specific algorithms. For couple of decades now, the use of DL classification techniques has shown to be highly performing in several fields of application such as image captioning, image classification, and speech recognition [26]–[28]. The possibility to extract complex information from a large amount of data is one of the first reasons to apply DL techniques in PSG classification.
The great advantage of the DL models is the high performance in dealing with a large amount of data. DL can learn features directly from the raw input data with little to no prior knowledge. However, the non-interpretability of the results and the longer computational times can be a drawback. On the other hand, features extracted starting from the knowledge of the experts are thought to be affected by several factors, primarily by the characteristics of the available dataset [29]. In sleep scoring the dataset present a wide variety, and the number of epochs in a single dataset is huge. The feature-based approach may not be suitable to satisfy a comprehensive description of the heterogeneity of the subjects and the set of recorded signals. For this reason, over the last few years, several works applied DL algorithms directly on raw PSG signals. In this short period of time, DL based algorithms have produced impactful results that were never seen with more conventional ML methods for a long time.

Convolutional Neural Networks and Recurrent Neural Networks

DL models are based on artificial neural networks and differ mainly on the architecture, which is how several neurons are arranged and connected to each other. Neurons lying on the same level make up the so-called *layer*. The network is composed of several units or neurons, each of them performs a linear combination of the input followed by a non linear transformation, as explained in details in Figure 2.2. The standard deep neural networks are characterized by multiple layers sequentially fully connected. Several types of DL architectures have been developed. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are the most widely used in ASSC.

A CNN is a supervised classification model in which the input (*e.g.*, raw data, spectrogram images) is processed by a network of filters and sub-sampling (pooling) layers. Each of these filters can be thought of as feature identifiers, whereas subsampling reduces the dimensionality but retains the important information. The last layer, usually a softmax layer, computes the output probability of each sleep stage to identify the target of the signal. An example of CNN overall architecture is

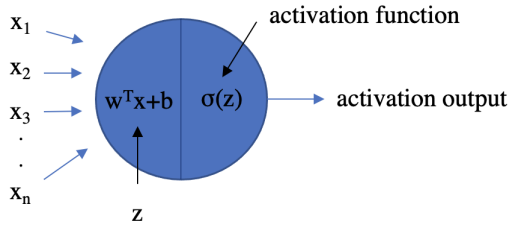


FIGURE 2.2: **A simple artificial neuron.** Each single neuron computes the dot product of the input $\boldsymbol{x}$ and the weight $\boldsymbol{w}$ vectors. A bias $\boldsymbol{b}$ value is added to the dot product and a non-linear function, called activation function (*e.g.*, sigmoid σ, hyperbolic tangent *tanh*, rectified linear unit or *ReLu*, leaky rectified linear unit), is then applied.

provided in Figure 2.3 and in Figure 2.4. During the training phase of the CNN the neuron filter weights and bias are adjusted in order to reach the target probability class for that input (epochs). After training, the CNN is ready to be applied on new input. As more layers are stacked more complex features are produced.

RNNs are networks of filters that can be trained, but they work on the principle of saving the output of a layer and feeding it back to the input, in order to predict the future output of the layer. CNN considers only the current input while RNN considers the current input together with the previously received inputs. Therefore, RNNs can easily handle sequential data.

In a sleep stage scoring procedure, the staging of each 30-second epoch is strongly related to the preceding and following epochs. Thus, over the years, a temporal dynamic behavior unit has been added to the CNNs - *i.e.*, the feature extractor or the epoch processing block - by introducing recurrent connections - *i.e.*, the sequence processing block. The more common memory units are a type of RNNs called long-short-term memory (LSTM) [30] and gated recurrent unit (GRU) [31]. This memory allows the model to process sequences of inputs (that are called epochs in our sleep staging task). That said, despite in literature many studies propose RNNs based architectures to encode the temporal dynamic behaviour, recently completely feed-forward architectures (*e.g.*, CNNs along with fully connected neural networks) have proven to be equally, if not even more, successful. This last statement will be further discussed in the next Chapter 3.

2.3 Sleep Databases

Several public and not public PSG datasets are employed to train DL based algorithms. The more common open access databases are Sleep-EDF and Sleep-EDF expanded version [32], followed by the Montreal Archive of Sleep Studies [33] and the Sleep Heart Health Study collection [34]. The biggest not public database belongs to the Massachusetts General Hospital Sleep Laboratory [35], that has 10000 recordings.

In Table 2.1 we report an extensive list of all the PSG datasets available online.

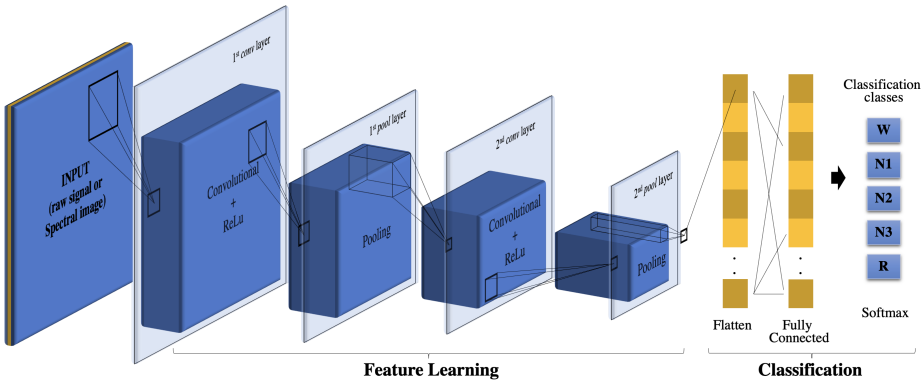


FIGURE 2.3: **CNN based sleep scoring architecture.** The CNN architecture can be divided in two subsequent parts, each performing a different process. The first, the feature learning activity, consists of several convolutional *conv* and of some pooling *pool* layers. The last, the classification process, is carried out by a fully connected layer and a softmax function. In the last part, the signal is flattened to one dimension, it is processed through a fully connected layer and finally classified using the last softmax layer. Unlike the *conv* layer, every unit of the fully connected layer interact with every input unit. The softmax layer computes the output probability of each sleep stage to identify the target of the signal.

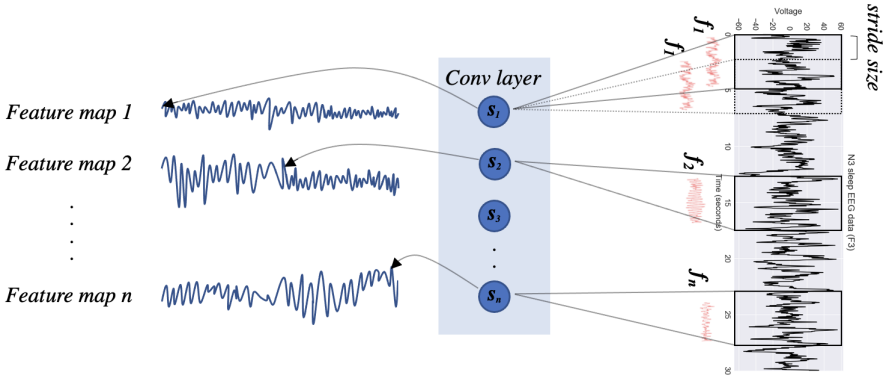


FIGURE 2.4: **1D-convolution operation.** Example of the 1D-convolution operation and feature maps construction in a *conv* layer. *Conv* layer firstly implements the convolution operation between the 30-second epoch and the n filters ***fi*** of the n neurons ***si***, then an activation function is applied (e.g *ReLu*). Each feature map *i* is the output of each convolution operation. Each value of a feature map can be considered as the result of the dot product between the local part of the input (size of the filter) and the filter ***f***. The dash-line window shows how each filter ***fi*** is shifted during the convolution (stride).

Datasets may differ for the sampling rate, the employed hardware - *i.e.*, differences in channels and derivations. Sometimes the subject category (*e.g.*, healthy or patients) as well as the human scorer identifier are missing. Clearly, an algorithm trained on healthy subjects would not keep the same performance on patients [52], even worse on patients with neurodegenerative diseases, *e.g.*, Parkinson's. The loss

of structure or function of neurons leads to important alteration of the EEG patterns.

As explained in section 2.1, in one night sleep, awake and N1 epochs are less frequent than others. As a consequence, PSG datasets are not balanced with respect to the number of classification targets (sleep stages). Usually there are a lot more epochs for N2 stage than for awake and N1 stages. Without balancing the datasets, it is highly likely that a classifier will exhibit skewed performance favoring the most represented classes, unless the least represented are very distinct from the other ones. In order to overcome this issue different approaches have been proposed: some authors apply *oversampling* on the sleep stages with fewer samples [29]; others apply *under-sampling* on the stages with a higher number of samples [53]. In some other studies, a weight is computed for each class, defined as the ratio between the frequency of the single class divided by the frequency of the most frequent class. The weights are assigned to each class as to contribute equally to the final prediction [54].

Last but not least, the datasets may have potential demographic and technical biases, *e.g.*, age, BMI, sex or hardware settings, that might significantly affect the learning and subsequentially the performance of the automatic scoring systems.

2.4 Deep Learning based Architectures

In the last decade, a wide variety of DL based architectures have been proposed for the sleep staging: autoencoders [55], deep neural networks (DNNs) [24], U-Net inspired architectures [56], [57], CNNs and fully-CNNs [7], [52], [53], [58]–[64], RNNs [65], [66] and several combination of them [29], [35], [54], [67]–[73]. The main advantage of these approaches is the ability to learn features directly from raw data, by also taking into account the temporal dependency among the sleep stages. The architectures of these models may be quite complex, a high number of parameters need to be trained - up to 24.7 million. The most recent ones process lengthy time sequences in input - up to 17.5 minutes. Most of them use RNNs, thus requiring extra resources to buffer the PSG input and making them unsuitable in home-monitoring and in real-time applications.

In most of these studies, CNNs and RNNs have been applied directly on raw PSGs data. Other approaches, that have shown performance on par if not better, are based on the usage of precomputed spectrograms (spectral images representing the frequency content of the signals over time) given in input to CNNs and RNNs based architectures [35], [52], [59], [71]. Recently, in [73], the combination of both raw PGS signals and spectrograms given in input to two parallel neural networks - *i.e.*, fully CNNs and attention-based RNNs - has proven to be highly efficient, outperforming state-of-the-art architectures on several databases. The main advantage of this latter approach is that it is capable to learn from both raw signal and the time-frequency images at the same time. The network is trained such that the learning pace on each input type is adapted based on their overfitting/generalization beahaviour.

One of the main goal of all the DL based architecture is to extract features from the data with a minimal manual pre-processing. Basic band pass filters (0.3-35Hz) are sometimes applied, as recommended in the AASM manual. During visual scoring, artifact removal is done using the contextual information. Unlike in feature-based approaches, only few DL algorithms consider the artifact reduction. Cui et al.

[61] use Butterworth filters to reduce some artifact from the signals, whilst Supratak et al. [29] apply a *weight decay* on the first layers of the CNNs in order to avoid overfitting to noises-artifacts in EEG data. In contrast, Malafeev et al. [54] point out that a DL algorithm can learn important information from epochs with artifacts, bringing as example the fact that wakefulness is almost always accompanied by movement artifacts and a movement is often followed by a transition into stage N1.

As a rule of thumb, deep architectures with a high number of layers and parameters need to be trained on large databases to prevent overfitting. Generally, increasing independent data-domains leads to an increase in the quality of the data analysis. In fact, in [57] the winning ingredient is the large amount of data (*i.e.*, hetereogeneity of the datasets and biases) used to train their U-Net inspired architecture, in combination with the heterogeneity of the EEG and EOG signals (*i.e.*, recording from different hardware) given in input to the algorithm.

There are still contrasting ideas and opinions regarding the channels and derivations to be used to train the scoring models. Clearly, their usage may also depend on the ultimate purpose of application, whether it is clinical or in home-monitoring. The vast majority of the existing architectures exploits information from different EEG channels. In particular Chambon et al. [53] have shown that the accuracy improves employing up to six well distributed EEG channels. They indicated that it is worth adding more EEG sensors, but up to a certain point. Biswal et al. [35] showed that there is a small reduction in performance between six and two EEG channel approach, but still on par with the level of accuracy attained by experts. It has also been shown that EEG together with EOG and EMG signal information leads to an increase in performance [73]. Hence, the usage of a multi-channel system improves the performance of the classification algorithms and can also better gather sleep information. However, sometimes the improvement is quite small whilst adding more channels can be computationally expensive, and could compromise the efficiency of the algorithm without leading to a far better classification. Many emerging home-based settings require a reliable solution with few channels. For these reasons, different groups have focused their attention on single-channel EEG or even on single-channel EOG analysis.

2.4.1 Benchmarking

The sleep scoring architectures, as commonly done for the ML algorithms, are evaluated splitting the data into a training set, a validation set and a test set. The training set is the database partition used to develop the algorithm, so high performance is expected. Validation and test sets are both independent from the data with which the model has been built. The validation set is used during the training phase, while the test set is used only to measure the final model performance. The validation procedure depends on the number of subjects available in the database. The data split commonly used in literature for databases with thousands of subjects (*e.g.*, SHHS, MROS, MESA) is 80% training and 20% test (hundreds of subjects held out from training set for validation set); another common data split choice is 75% training, 10% validation and 15% test. When there are less than a few hundred subjects in the database, the performance is evaluated with a cross-fold validation procedure, using both k-fold and leave-one-out methodologies. In k-fold cross-validation, the PSG dataset is randomly partitioned into k equal sized recording groups. Out of the k groups a single group is retained as the test set data for testing the model;

the remaining k-1 subgroups are used as training data, and further split into a validation set (*e.g.*, 10% of the training set) for evaluating the model during training. The cross-validation process is then repeated k times, with each of the k groups used exactly once as the test data. The k results can then be averaged to produce a single estimation. The advantage of this method is that all observations are used for training/validation/test, and each observation is used for testing exactly once. When k = n (the number of observations), the k-fold cross-validation is exactly the leave-one-out cross-validation.

The results are represented generally with the percentage agreement between the classifier and the gold standard, that is the visual scoring by a human expert. The performance of the scoring algorithms are usually evaluated using the macro-averaging F1-score (MF1), the per-class F1-score, the overall accuracy (Acc.), the Cohen's kappa (k), the average sensitivity and the average specificity. The F1-score, sensitivity and specificity are given for each sleep stage, and resemble in general the same problematics as in the visual scoring procedure. N1 is the more difficult stage to be identified. The Cohen's kappa is generally thought to be a more robust measure as it takes into account the possibility of the agreement occurring by chance.

The evaluation metrics are calculated considering the human visual scoring as the gold standard. Consequentially, a performance similar to the inter-scorer agreement (that is on average around 80%) should be considered an excellent result, whilst higher performance may be considered, in most of the cases, as overfitting on the dataset. The information of the training set and of the test set belongs to different recordings, but if they still belong to the same dataset they cannot be considered totally independent. A dataset usually comes from the same sleep center and contains recordings from the same expert scorers. The high percentages of accuracy could be the result of an overfitting phenomenon (the models fit to the specific dataset). Data from different sleep labs and data from several cohorts ensure the reproducibility of the developed methods. Some authors have measured the performance of their model using test set coming from external database [35], [52], [57], [67]; their results should be then considered more robust.

Keeping in mind all the previous considerations, it appears quite clear the great difficulty of comparing different author works. They evaluate their architectures on different database, by using different derivations-hardware, different amount of subjects and metrics. Moreover, even if the architectures have been evaluated on the same database, the validation procedures are not comparable. In Supplementary Table B.1 we report an up-to-date overview of the available DL based scoring architectures, along with the dataset characteristics, subject type (healthy or unhealthy), information sources (channels), DL network type (classifier) and performance. Most of them reach very good performance in terms of overall accuracy, compared with the inter-scorer agreement. In order to decide which classifier is better than the others, all the classifiers should be developed using the same channels, trained on the same dataset and validated with the same procedure.

2.5 Automated Sleep Scoring in Clinical Practice

In light of all the recent well-performing (*i.e.*, intra-scoring agreement accuracy level) scoring architectures, one question arises: why these automated scoring algorithms

are not already routinely adopted in all the sleep centers? In the following points the perspectives of ICT researchers and sleep scoring experts are summarized.

- ***Aversion to technology***: in the health care domain, new technologies are perceived often as a threat [74], especially if these new tools are going to substitute part of the work done by human beings and if they intervene somehow in the diagnosis.
- ***Usability***: many tools that are on the market are not easy to use and have not a friendly user interface [75] ; a user-centered design should be favored [76].
- ***Security and privacy issues***: some powerful scoring service requires the uploading of the sleep recordings data to the cloud, *i.e.*, Z3Score [77], or externally to secure servers, *i.e.*, Michele [78]. This action is often forbidden or discouraged inside the hospitals [79].
- ***Dataset biases***: automatic scoring works well on healthy subjects. The majority of the ML approaches for improving sleep scoring have used a training set of healthy adult male subjects. Consequentially applying these algorithms to patients with sleep disorders [17], [54] or neurodegenerative disorders [52], [80] often fails.
- ***Scoring rules***: the actual scoring rules leave space for subjective interpretation, leading to a high inter- and intra-scorer variability. Moreover the rules, based on 30-second epochs, tend to consider sleep stages as distinct entities, while sleep should be viewed as a gradual transition from a stage to the other [81]. Younes et al. [82] in a recent paper state that data interpretation performed by only one technologist should be considered unreliable.

TABLE 2.1: **Online databases overview.**

Datasets	Recordings	Subjects	Age (years)	Sex % F/M
ABC (✓)	132	49	48.8±9.8	43/57
CCSHS (✓)	515	515	17.7±0.4	50/50
CFS (✓)	730	730	41.7±20.0	55/45
CHAT (✓)	1638	1232	6.6±1.4	52/48
DCSM ✓	255	255	-	-
DOD-H ✓	25	25	35.3±7.5	24/76
DOD-O ✓	55	55	45.6±16.5	36/64
ISRUC-SG1 ✓	100	100	51.1±15.9	44/56
ISRUC-SG2 ✓	16	8	46.9±17.5	25/75
ISRUC-SG3 ✓	10	10	39.6±9.6	10/90
HPAP (✓)	238	238	46.5±11.9	43/57
MASS-C1 (✓)	53	53	63.6±5.3	36/64
MASS-C3 (✓)	62	62	42.5±18.9	55/45
MESA (✓)	2056	2056	69.4±9.1	54/46
MROS (✓)	3926	2903	76.4±5.5	0/100
PHYS ✓	994	994	55.2±14.3	33/67
SEDF-SC ✓	153	78	58.8±22.0	53/47
SEDF-ST ✓	44	22	40.2±17.7	68/32
SHHS (✓)	8444	5797	63.1±11.2	52/48
SOF (✓)	453	453	82.8±3.1	100/0
SVUH-UCD ✓	25	25	50.0±9.4	16/84

ABC: The Apnea, Bariatric surgery, and CPAP [36], [37]; CCSHS: The Cleveland Children's Sleep and Health Study [36], [38]; CFS: The Cleveland Family Study [36], [39]; CHAT: The Childhood Adenotonsillectomy Trial [36], [40], [41]; DCSM: The Danish Centre for Sleep Medicine; DOD-H & DOD-O The Dreem Open Dataset – Healthy & Obstructive [25], [42]; ISRUC: The Sleep Medicine Centre of the Hospital of Coimbra University [43]; HPAP: The Home Positive Airway Pressure [36], [44]; MASS: The Montreal Archive of Sleep Studies [33]; MESA: The Multi-Ethnic Study of Atherosclerosis [36], [45]; MROS: Study of Osteoporotic Fractures in Men [36], [46], [47]; PHYS: 2018 PhysioNet/CinC Challenge by the Massachusetts General Hospital's Computational Clinical Neurophysiology Laboratory and the Clinical Data Animation Laboratory [48], [49]; SEDF: The Sleep-EDF Database (Expanded) [32], [48]; SHHS: The Sleep Heart Health Study [34], [36]; SOF: Study of Osteoporotic Fractures [36], [50], [51]; SVUH-UCD: The St. Vincent's University Hospital / University College Dublin Sleep Apnea Database [48].
Datasets directly available online are identified by ✓, whilst datasets that require approval from a Data Access Committee marked by (✓).

Chapter 3

Automated sleep scoring, temporal dependency and recurrent neural networks

In Chapter 2 we have provided a general overview of the latest DL based architectures proposed to automatize the sleep scoring procedure. These models can be grouped into four classification schemes (see Figure 3.1), based on the number of epochs in input and on the number of sleep stages (*i.e.*, sleep labels) in output:

(**a**) **one-to-one or epoch-to-epoch**. It is the simplest classification scheme. The architecture receives in input a single PSG epoch, and it outputs a single corresponding sleep stage [83]. Formally, given in input the epoch $\mathbf{x}_t$, the epoch-to-epoch sleep staging approach is designed to maximize the conditional probability $P(y_t|\mathbf{x}_t)$, where y_t is the $t-th$ one-hot encoded vector of the ground-truth label. The clear drawback is that this classification scheme does not take into account the temporal dependency that exists between the epochs.

(**b**) **many-to-one or sequence-to-epoch**. The architecture receives in input a sequence of PSG epochs, and it outputs a single sleep stage corresponding to the central epoch in the sequence [52]. Formally, given in input the sequence of L epochs $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_L)$, the *many-to-epoch* sleep staging approach is designed to maximize the conditional probability $P(y_t|\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_L)$, where y_t is the one-hot encoded vector of the ground-truth label corresponding to the central epoch $\mathbf{x}_t$ in the sequence. In that case a contextual input is given to the architecture, actually emulating what the physician does during the manual scoring procedure. This classification scheme was the most popular until 2018.

(**c**) **one-to-many or epoch-to-sequence**. It is the orthogonal scheme to the commonly used many-to-one classification scheme. The architecture receives in input a single PSG epoch, and it outputs simultaneously its sleep stage and the sleep stages in its neighbourhood (*i.e.*, the contexual output). Formally, given in input the central epoch $\mathbf{x}_t$ in a sequence L, the *epoch-to-many* sleep staging approach is designed to maximize the conditional probability $P(y_1, y_2, ..., y_L|\mathbf{x}_t)$, where $(y_1, y_2, ..., y_L)$ is the sequence of the corresponding L one-hot encoded vectors of the ground-truth label. The rational behind this approach, proposed in [84], is that given the temporal dependency between the PSG epochs, we should be able to predict the sleep stages of the neighbors only using the information of a single epoch.

(**d**) **many-to-many or sequence-to-sequence**. The architecture receives in input a sequence of PSG epochs, and it outputs the corresponding sequences

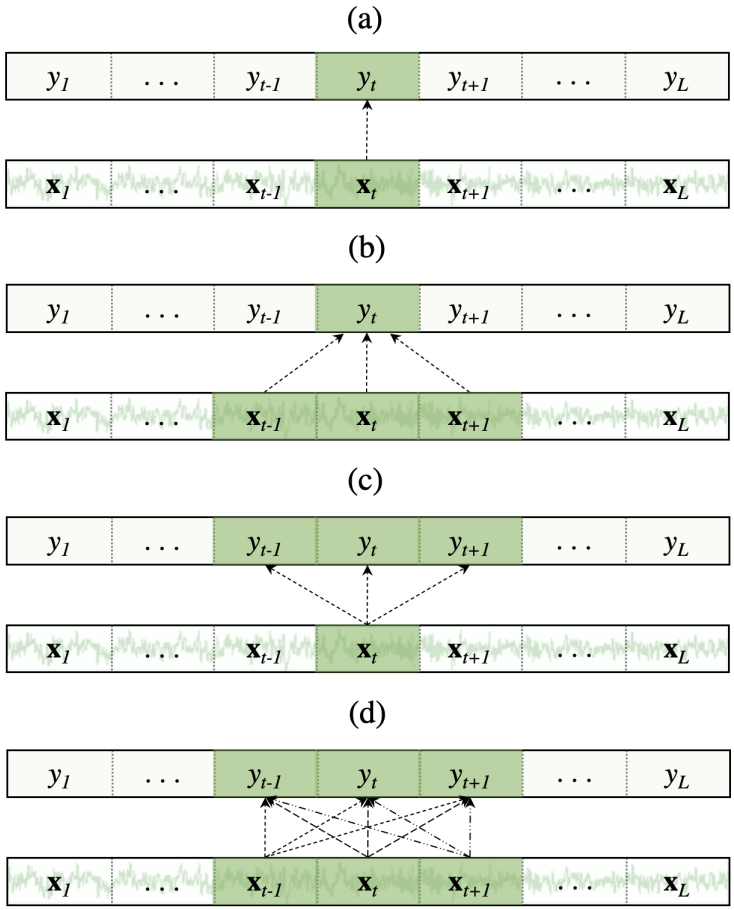


FIGURE 3.1: **Classification schemes for automatic sleep scoring.** Classification schemes for automatic sleep scoring: (a) one-to-one or epoch-to-epoch, (b) many-to-one or sequence-to-epoch, (c) one-to-many or epoch-to-sequence, and (d) many-to-many or sequence-to-sequence.

of sleep stages at once [66]. Formally, given in input the sequence of L epochs $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_L)$, the many-to-many sleep staging approach is designed to maximize the conditional probability $P(y_1, y_2, ..., y_L | \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_L)$, where $(y_1, y_2, ..., y_L)$ is the sequence of the corresponding L one-hot encoded vectors of the ground-truth label. The majority of the recently proposed sleep scoring architectures follow this classification scheme. The main advantage is that it exploit both the contextual input and the contextual output at once.

In the last five years, the common trend of the many-to-many based scoring architecture was to divide the framework in three main blocks (see Figue 3.2): the epoch processing block (*EPB*), the sequence processing block (*SPB*) and the classification block (often a simple fully connected layer followed by a softmax function).

- The *EPB* is an epoch-wise feature learner block. It takes in input an epoch $\mathbf{x}_t$ (*e.g.*, the raw signal or the time-frequency image) in a sequence L, where $1 \leq t \leq L$. The input can be a single-channel or a combination of multiple channels. The *EPB* is applied independently on all the epochs in the sequence in input, and it transforms each $\mathbf{x}_t$ into a feature representation vector $\mathbf{f}_t$.

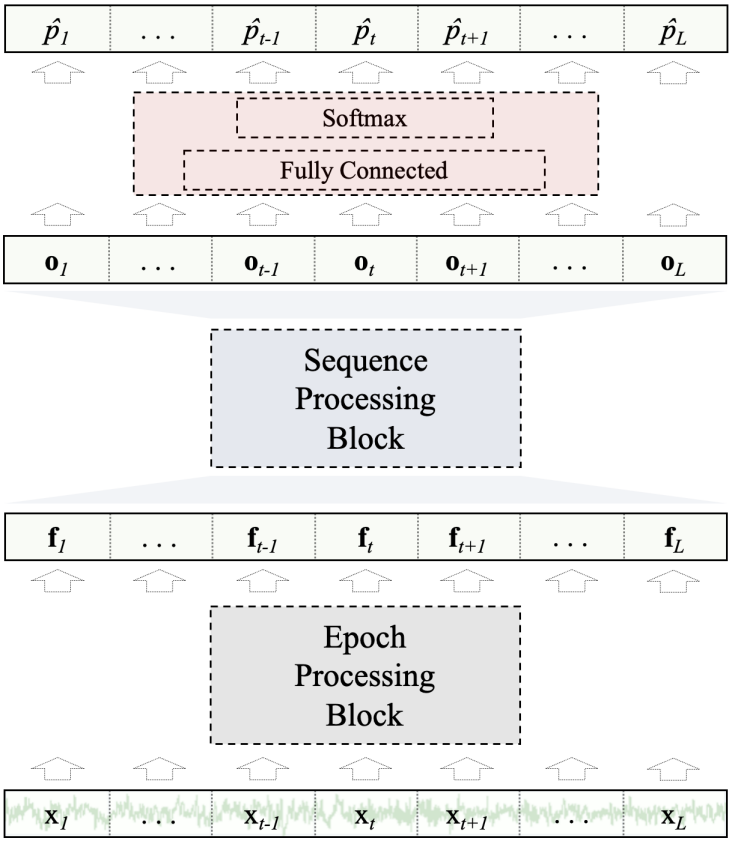


FIGURE 3.2: **Framework of the sleep scoring architectures.** The framework of the sleep scoring architecture is divided in three blocks: the epoch processing block (*EPB* in grey), the sequence processing block (*SPB* in light blue) and the classification block (fully connected layer and the softmax function in light red).

- The *SPB* block encodes the sequence of the epoch-wise feature vectors from *EPB* $(\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_L)$ into the sequence of output vectors $(\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_L)$. The purpose of this block is to encode the temporal dependency between the epochs of the sequence L.
- The classification block exploit a fully connected (FC) layer along with the softmax activation function to output the sleep stage probabilities $(\hat{p}_1, \hat{p}_2, ..., \hat{p}_L)$, *i.e.*, the final predictions, from the sequence of output vectors $(\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_L)$.

The idea to use a sequence processing block in cascade to an epoch processing block comes from the need to capture the long-range dependencies between the sleep epochs. Supratak et al. [29] were the first to add two layers of LSTM to encode a long sequence of epoch-wise feature vectors, $(\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_L)$, into a sequence of output vectors, $(\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_L)$, before the softmax layer. DeepSleepNet significantly boosted the automated scoring performance at that time. Inspired by their results, the community started to exploit the *SPB* block, by using RNNs based architectures. The aim was to enhance the long-term capacity of the classification model.

In Chapter 3 we describe experiments performed starting from the well-known DeepSleepNet [29] architecture, state-of-the-art architecture at the time. Our main

assumption was that the sequence processing blocks (*i.e.*, the RNN layers) were not effectively encoding the temporal dependencies between the epochs within the sequence. The reason why the architectures succeed in encoding the temporal dynamic behaviour may not necessarily relate to the recurrent blocks, but to the temporal context they were giving in input. Thus, we propose to replace the recurrent layers with simple FC layers, or to directly remove the RNN layers and solve the classification task by only using the CNN layers. We demonstrate that the long-range dependencies between the sleep epochs can be encoded quite well by also using a simple feedforward architecture.

Contributions. Our contributions can be summarized as follows: (1) we prove how simple feed forward architectures achieve comparable performance to the recurrent neural network based ones; (2) we show that a feed forward architecture, trained with a small temporal context (*i.e.*, three sleep epochs), already achieves quite good performance; (3) we attempt to introduce a new metric to better evaluate the ability of each network to model sequential information.

3.1 Architectures

To better understand all the experiments, the different training approaches and the models proposed in this chapter, it is worth first spending few words about the architecture originally proposed by Supratak et al. [29].

DeepSleepNet consists of two main parts as shown in Figure 3.3.

- The *representation learning* part, or what we refer to as *EPB*, is designed to process 30-second single-channel EEG epochs, and it aims at learning epoch-wise features. It consists of two parallel *CNNs* employing small (CNN_{θ_S}) and large (CNN_{θ_L}) filters at the first layer. The small filter has been used to extract high-time resolution patterns, while the large filter has been used to extract high-frequency resolution patterns. The idea behind the use of the small and large filter sizes comes from the way the signal processing experts define the trade-off between temporal and frequency precision in the feature extraction procedure [85]. Each *CNN* section consists of four convolutional layers and two max-pooling layers. Each convolutional layer executes three operations: a one-dimensional convolution of the filters with the 30-second EEG epochs, a batch normalization [86] and an element-wise rectified linear unit (ReLU) activation function. The pooling layer is used to downsample the input. The filters size, the number of filters, the stride size of each *conv* layer, the pooling size and the stride size of the pooling layers are all defined in Figure 3.3.
  Each 30-second EEG epoch $\mathbf{x}_i$ is given in input to the convolutional neural networks CNN_{θ_S} and CNN_{θ_L}. The parameters θ of each CNN are independently trained, so as to return in output two feature vectors $\mathbf{h}_i{}^S$ and $\mathbf{h}_i{}^L$. The outputs are concatenated in $\mathbf{f}_i$, then forwarded to the *sequence residual learning* part.

$$\mathbf{h}_i{}^S = CNN_{\theta_S}(\mathbf{x}_i) \tag{3.1}$$

$$\mathbf{h}_i{}^L = CNN_{\theta_L}(\mathbf{x}_i) \tag{3.2}$$

$$\mathbf{f}_i = \mathbf{h}_i{}^S || \mathbf{h}_i{}^L \tag{3.3}$$

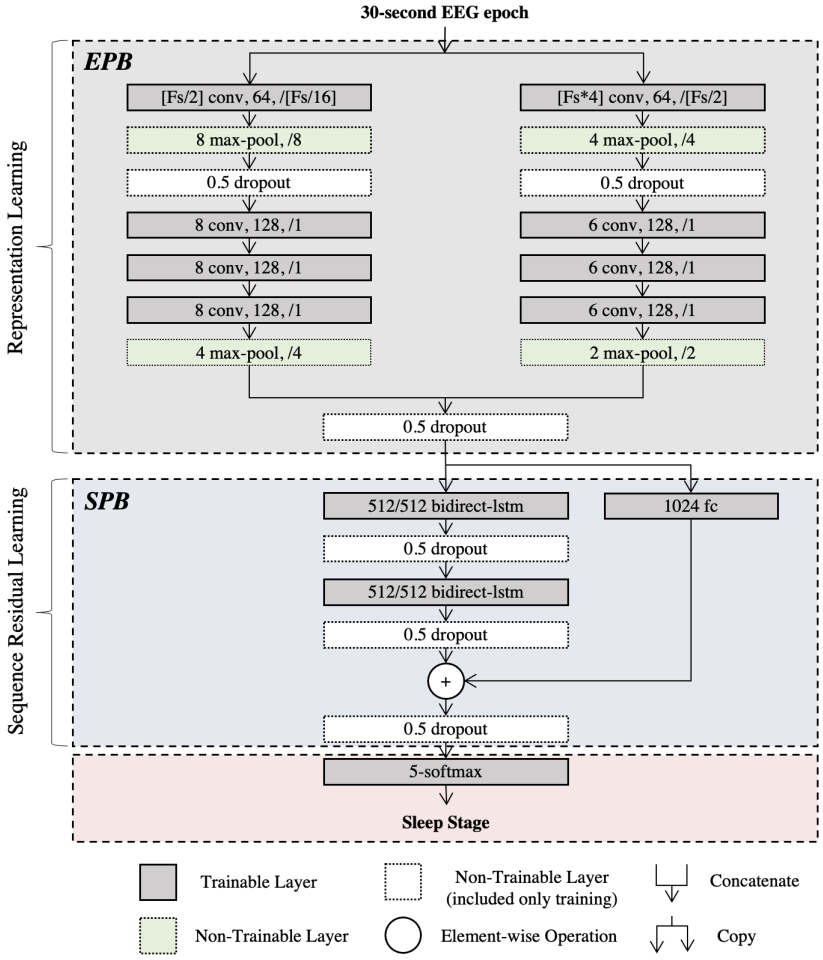


FIGURE 3.3: **DeepSleepNet architecture.** An overview of DeepSleepNet architecture consisting of two main parts: representation learning (*EPB*) and sequence residual learning (*SPB*). Each trainable layer is a layer containing parameters to be optimized during a training process.

- The *sequence residual learning* part, or what we refer to as *SPB*, is designed to process sequences of epochs, and it aims to encode the temporal information (*e.g.*, stage transition rules) from the sequence of epoch-wise feature vectors $\mathbf{f}_i$. It consists of two layers of bidirectional-LSTM [87] and a shortcut connection. Th bi-LSTM layer differs from the standard LSTM layer by having two LSTMs process forward f and backward b input sequences independently.
  Given the features $(\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_L)$ from the *CNNs*, where $1 \leq t \leq L$ denotes the time index of the 30-second EEG epochs, the sequence residual learning block operates as follows:

$$\mathbf{h}_t^f, \mathbf{c}_t^f = LSTM_{\theta_f}(\mathbf{h}_{t-1}^f, \mathbf{c}_{t-1}^f, \mathbf{f}_t) \tag{3.4}$$

$$\mathbf{h}_t^b, \mathbf{c}_t^b = LSTM_{\theta_b}(\mathbf{h}_{t+1}^b, \mathbf{c}_{t+1}^b, \mathbf{f}_t) \tag{3.5}$$

$$\mathbf{o}_t = \mathbf{h}_t^f || \mathbf{h}_t^b + FC_\theta(\mathbf{f}_t) \tag{3.6}$$

LSTM process the sequences of features $\mathbf{f}_t$ with the two-layers of LSTM, parameterized by θ_f for forward and by θ_b for backward directions; $\mathbf{h}$ and $\mathbf{c}$ represent the vectors of the hidden and cell states of the LSTMs; *FC* transforms $\mathbf{f}_t$ into a vector to be added to the concatenated $\mathbf{h}_t{}^f$ and $\mathbf{h}_t{}^b$ vectors. The hidden size for the forward and backward LSTMs, along with the size of the *FC* layer are defined in Figure 3.3.

The softmax function, together with the cross-entropy loss function H, is used to train the model to output the logits $\mathbf{z}_i$ and the probability for the five mutually exclusive classes that correspond to the five sleep stages.

$$\mathbf{z}_i = \mathbf{W}^T \mathbf{o}_i + \mathbf{b} \tag{3.7}$$

$$\hat{p}_{i,k} = \frac{exp(z_{i,k})}{\Sigma_j exp(z_{i,j})} \tag{3.8}$$

$$H(\mathbf{y}_i, \mathbf{p}_i) = \sum_{k=1}^{K} -y_{i,k} \cdot \log(\hat{p}_{i,k}) \tag{3.9}$$

where θ = {**W**,**b**} are the parameters of the softmax layer, i is the $i-th$ 30-second EEG epoch $\mathbf{x}_i$ in the sequence of lenght L; $\mathbf{o}_i$ is the output of the *sequence residual learning* part associated to $\mathbf{x}_i$; j is the index of the vector $\mathbf{z}$; $\hat{p}_{i,k}$ is the output probability of class k associated to $\mathbf{x}_i$.

DeepSleepNet is trained end-to-end via backpropagation in two steps. In the first step the *representation learning* part is pre-trained epoch by epoch, *i.e.*, epoch-to-epoch classification scheme. Note that the two *CNNs* are stacked with a softmax layer (which we will refer to as softmax*). This softmax* layer is only used in this step to pre-train the two CNNs; its parameters are discarded at the end of the pre-training step. In the second step, the whole architecture (the *sequence residual learning* in cascade to the *representation learning*) is fine-tuned sequence by sequence, *i.e.*, sequence-to-sequence classification scheme. For a detailed description about the network, the two-step training algorithm and the training parameters we refer the reader to [29].

In the following we propose two architectures that differ from [29] mainly in the second learning block *SPB*. They are all trained in two step, but the fine-tuning is performed using both a sequence-to-sequence classification scheme and a sequence-to-epoch classification scheme, for which we will use the corresponding names ***SeqToSeq*** and ***SeqToEpoch***. In Supplementary Figure B.1 we report the classification scheme of the original DeepSleepNet architecture.

bi-LSTM. The first *EPB* block is as in [29], whilst, instead of the two bidirectional-LSTM layers along with the shortcut connection FC, only a single layer of bidirectional-LSTM cells is used in the *SPB* block. Unlike the original network, the input of the sequential part is the sequence of the L vectors of $\mathbf{z}^*{}_t$ logits stacked; where L is the number of 30-second PSG epochs considered in a sequence, and the $\mathbf{z}^*{}_t$ logits are the output of the first softmax* (in which its parameters are not discarded) used during the pre-training of the first *representation learning* part. The fine-tuning of the whole architecture is performed with a sequence-to-sequence classification scheme, resulting in ***SeqToSeq-bi-LSTM*** (see Supplementary Figure B.2), and with a sequence-to-epoch classification scheme, resulting in ***SeqToEpoch-bi-LSTM*** (see

Supplementary Figure B.3).

FFNN (*Feed Forward Neural Networks*). The first *EPB* block is as in [29], whilst the *SPB* consists of two fully connected layers, with ReLU non-linearity and batch normalization. The FC layers have 500 and 250 hidden units respectively. The input of the sequential part is the sequence of the L vectors of $\mathbf{z}^*_t$ logits stacked, as in the previous architecture. The fine-tuning of the whole architecture is performed with a sequence-to-sequence classification scheme, resulting in ***SeqToSeq-FFNN*** (see Supplementary Figure B.4), and with a sequence-to-epoch classification scheme, resulting in ***SeqToEpoch-FFNN*** (see Supplementary Figure B.5).

All the architectures share the same pre-trained *EPB* block proposed in [29]. It receives as input the single PSG epoch $\mathbf{x}_t$ and output the corresponding single sleep stage y_t. We refer to this simple architecture as ***EpochToEpoch-EPB*** (see Supplementary Figure B.6), as the pre-training of the first block is done using the simplest epoch-to-epoch classification scheme. In our experiments we have also trained the *EPB* block using the sequence-to-epoch classification scheme, resulting in ***SeqToEpoch-SPB*** (see Supplementary Figure B.7). Thus, the architecture receives the contextual input of an epoch and the epoch itself, and it predicts the corresponding target of the centred 30-second signal. In this case, the *EPB* block has to be considered as a *SPB* block, as it processes the whole sequence/signal instead of only one 30-second EEG epoch at a time.

As in [29], in all the experiments, we adopted a sequence of length $L = 25$, unless otherwise specified. All the training parameters and the regularization techniques are identical as in DeepSleepNet. The architecture has several hyperparameters (*e.g.*, number of layers, number/sizes of filters, regularization parameters, training parameters etc.) which could be optimized to tune its performance on any dataset. We decided to not systematically tune all these parameters - out of our scope - but to fix them for all the experiments, as done in the original network.

3.2 Database

SEDF-SC-13. The Sleep-EDF Sleep Cassette, is a subset of the open source Sleep-EDF Expanded dataset [32], [48]. In order to facilitate the comparison with the original *DeepSleepNet*, we use the previous upload of the Sleep-EDF database, published in 2013 (to which we will refer as SEDF-SC-13). The database contains PSGs from 20 subjects (10 males and 10 females) aged from 25 to 34, sampled at 100 Hz. Except for the second night of the subject 13, for all the subjects are available two whole nights, resulting in 39 PSG recordings. Each recording includes two scalp EEG channels (Fpz-Cz and Pz-Cz), one EOG (horizontal) channel, one submental chin EMG channel and one oro-nasal respiration channel. The recordings are manually scored by sleep experts on 30-second epochs according to R&K scoring rules [9], resulting in the eight classes Wake, N1, N2, N3, N4, REM, MOVEMENT and UNKNOWN. In order to use the AASM standard [2], we have merged the N3 and N4 stages into a single stage N3, and we have excluded the MOVEMENT and UNKNOWN classes. In many recordings there were long wake periods before the patients went to sleep and after they woke up. As in [29], only 30 minutes of data before and after *in-bed* parts have been taken into account. In our experiments we have considered the

single-channel EEG Fpz-Cz, with a sampling rate of 100 Hz and without any pre-processing.

3.3 Results

3.3.1 Experiment Designs

We evaluate each model using the k-fold cross-validation scheme. We set k equal to 20, *i.e.*, the total number of subject in the database. We used in each fold recordings from 19 subjects to train the models, and the recordings from the left out subject to test the trained model (*i.e.*, leave-one-out cross validation scheme). This process is repeated k times so that all of the recordings are tested.

3.3.2 Metrics

The per-class F1-score, the overall accuracy (*Acc.*), the macro-averaging F1-score (*MF1*) and the Cohen's kappa (k) have been computed from the predicted sleep stages from all the folds to evaluate the performance of each model [88], [89].

Temporal Encoding. In order to better evaluate the ability of each network to model sequential information, we attempt to introduce the new metric δ_{norm}.
We noticed that misclassifications usually appear in the sleep-phase-transition proximity; the networks show clear difficulties in codifying the sequential information corresponding to transitions. It was also clear that when there is a misclassification the second highest probability value was almost always the correct one. Indeed the performance of all the architectures in a Top-2 classification increases in overall accuracy, on average above 92%. It can be assumed that the difference between the first max logit value and the second max logit value decreases if there is a misclassification, and increases if there is not.
The output of the last softmax layer is $\hat{p}_{\mathrm{i,k}}$, *i.e.*, the output probability for each class k associated to $\mathbf{x}_i$ (eq 3.9). We define the δ parameter as the ratio between $\Delta_{\mu,incorrect}$ and $\Delta_{\mu,correct}$.

$$\Delta_{\mu,\mathrm{incorrect}} = \frac{\sum_{i=1}^{n^*_{\mathrm{incorrect}}} (\max(\hat{p}_{\mathrm{i}}) - 2^{\mathrm{nd}} \max(\hat{p}_{\mathrm{i}}))}{i} \tag{3.10}$$

$$\Delta_{\mu,\mathrm{correct}} = \frac{\sum_{i=1}^{n_{\mathrm{correct}}} (\max(\hat{p}_{\mathrm{i}}) - 2^{\mathrm{nd}} \max(\hat{p}_{\mathrm{i}}))}{i} \tag{3.11}$$

$$\delta = \frac{\Delta_{\mu,\mathrm{incorrect}}}{\Delta_{\mu,\mathrm{correct}}} \tag{3.12}$$

$\Delta_{\mu,incorrect}$ is the averaged value of the differences between the max probability and the second max probability computed on all the misclassified ($n^*_{\mathrm{incorrect}}$) epochs where the second max probability matched the true target. Whilst, $\Delta_{\mu,correct}$ is the averaged value of the differences between the max probability and the second max probability computed on all the correctly classified (n_{correct}) epochs. In our analysis we have considered the normalized parameter δ_{norm}.

$$\delta_{norm} = \frac{\Delta_{\mu,\mathrm{incorrect}}}{\Delta_{\mu,\mathrm{correct}}} * \frac{\frac{n^*_{\mathrm{incorrect}}}{n_{\mathrm{incorrect}}}}{\frac{n_{\mathrm{correct}}}{n_{\mathrm{total}}}} \tag{3.13}$$

TABLE 3.1: **Overall performance on all the experiments on SEDF-SC-13.** Training parameters, overall performances, per-class F1-score and δ_{norm} parameter of each architecture obtained from 20-fold cross-validation, with best shown in bold.

Methods	Training Param.	Overall Metrics			Per-class F1-Score					δ_{norm}
		Acc.	MF1	k	W	N1	N2	N3	REM	
DeepSleepNet	$\sim 24.7M$	82.0%	76.9%	0.76	84.7%	46.6%	85.9%	84.8%	82.4%	0.73
SeqToSeq-bi-LSTM	$\sim 1.1M$	**85.2%**	**79.9%**	**0.80**	**88.5%**	50.2%	88.4%	86.1%	**86.3%**	0.47
SeqToSeq-FFNN	$\sim 0.8M$	82.7%	76.0%	0.76	87.5%	41.4%	86.4%	81.8%	82.9%	**0.43**
SeqToEpoch-bi-LSTM	$\sim 1.1M$	84.9%	**79.9%**	0.79	87.2%	49.2%	**88.6%**	**88.6%**	86.2%	0.56
SeqToEpoch-FFNN	$\sim 0.8M$	84.0%	79.1%	0.78	85.1%	**50.9%**	88.3%	86.9%	84.2%	0.49
EpochToEpoch-EPB	$\sim 0.6M$	81.1%	75.0%	0.75	85.8%	38.5%	87.1%	86.8%	76.8%	0.51
SeqToEpoch-SPB	$\sim 1.0M$	83.1%	77.3%	0.77	85.4%	44.4%	87.3%	88.1%	81.2%	0.63
3-SeqToEpoch-SPB*	$\sim 0.6M$	83.9%	78.9%	0.78	87.4%	49.1%	87.7%	88.3%	82.0%	0.55

where $n_{\text{incorrect}}$ is the number of epochs misclassified, $n^*_{\text{incorrect}}$ is the number of epochs misclassified with the second max probability matching the true target, n_{correct} is the number of epochs well classified and n_{total} is the total number of classified epochs.

The δ parameter lies in the interval $]0, \infty[$, where the tendency to zero value were supposed to denote a good sequential soft classifier, and the tendency to ∞ value were suposed to denote a bad sequential soft classifier. In the next subsection 3.3.3 we will explain the real meaning of this δ parameter, and its close relationship with the concept of calibrated neural networks.

3.3.3 Analysis of Experiments

In Table 3.1 we report the performances of each architecture, along with their complexity - *i.e.*, the number of parameters to be trained. The overall metrics and the per-class F1-score are computed for each model over the 20-fold cross-validation.

SeqToSeq-bi-LSTM and *SeqToEpoch-bi-LSTM*, that are less complex versions of the original DeepSleeNet (*i.e.*, number of training parameters reduced by a factor of 22.5), achieve the best results among all the networks. The *SeqToSeq* classification scheme reaches an overall accuracy of 85.2% and k equal to 0.80, whilst the *SeqToEpoch* scheme reaches an overall accuracy of 84.9% and k equal to 0.79.
SeqToSeq-FFNN and *SeqToEpoch-FFNN* have slightly lower performance, even though the number of parameters is about half the number of parameters for the *bi-LSTM* based models. The feed forward architecture along with the *SeqToEpoch* classification scheme reaches an overall accuracy of 84.0% and k equal to 0.78. The results obtained by using only the first block, trained with the *SeqToEpoch* classification scheme and a temporal context in input of only 3 epochs, are quite interesting. The *3-SeqToEpoch-SPB* model reaches an overall accuracy of 83.9% and k equal to 0.78.

3.4 Discussion

This first preliminary study sets the stage for the experiments we will go through in the next Chapter 4, with the ultimate goal of quantifying the disagreement between the predictions given a sleep scoring algorithm and the physician's annotations. Here, we propose to simplify DeepSleepNet, an existing state-of-the-art RNN based sleep scoring architecture. We prove how simpler feed forward architectures achieve comparable performance to that of a recurrent neural network approach, on a small-sized dataset (*i.e.*, low heterogeneity between subjects). The reason why the architectures succeed in encoding the temporal dynamic behaviour (*e.g.*, sleep stage transitions) may not necessarily relate to the recurrent blocks, but to the temporal context we give as input. From these results, we can assume that the first part of the network, *i.e.*, the CNN based processing block, does most of the work. However, there is still a gap to fill to reach a higher level of performance. This gap can be filled with information related to the sequentiality, mimicking the human visual analysis procedure (see the architecture exploited in Chapter 6.

As a result of follow-up analysis, we realized that we cannot rely on the δ_{norm} parameter to evaluate the ability of each method to model the sequential information (*i.e.*, the temporal dependency between the sleep epochs). Rather, this parameter returns incomplete information about the calibration of each model. By definition, a model is perfectly calibrated when the probability associated to the predicted stage mirrors its ground truth correctness likelihood. So far, different metrics have been proposed to evaluate the calibration of a model (*e.g.*, *ECE* [90]). In our approach, we have practically quantified a similar information, but computing the difference between the *max* probability values and the second *max* probability values in output from our networks. In the next Chapter 4 we will further explore this topic, we will give more details about the calibration of a model, we will provide a potential solution to be able to quantify the uncertainty of a sleep scoring model.

Chapter 4

A simplified sleep scoring model with uncertainty estimates

Most of the architectures we mention in Supplementary Table B.1 are quite complex (*i.e.*, high number of parameters to be trained and lengthy time sequences to process - up to 12 minutes). As a rule, deep architectures with a high number of layers and parameters need to be trained on large databases to prevent overfitting. In different scenarios sleep datasets have a limited number of labeled PSG samples available. Lighter architectures may be better suited if the model needs to be trained from scratch. Besides, only two architectures [52], [67] perform the automatic sleep scoring also providing an estimate of the model uncertainty. In [52] they use an additional classification block-2 (*i.e.*, multilayer perceptron in cascade to the deep convolutional scoring architecture) to output the final sleep stage and the associated relative confidence score. In contrast, [67] trains 16 different models and uses the relative model variance to estimate the uncertain predictions. The common purpose and our ultimate perspective is to quantify the level of confidence for each prediction, as it could be the key to identify the misclassified sleep stages, to then send them back to the physician for a secondary review, *i.e.*, a closed-loop interaction between the algorithm and the end-user.

In Chapter 4 we introduce DeepSleepNet-Lite (DSN-L), a simplified and lightweight automatic sleep scoring architecture. It provides the predicted sleep stages along with an estimate of their uncertainty. The major advantage is that it does not require any additional computation over the baseline architecture to provide this estimate.

Contributions. Our contributions can be summarized as follows: (1) we show that DSN-L achieves performance on par with most up-to-date scoring systems; (2) we demonstrate the efficiency of label smoothing and Monte Carlo dropout sampling techniques in both calibrating and enhancing the performance of our model; (3) we propose a new conditional probability distribution, computed over the targets (*i.e.*, our prior knowledge), to smooth the labels; (4) we prove the efficiency of our uncertainty estimate procedure, by showing that it is able to identify the most challenging sleep stage predictions.

4.1 DeepSleepNet-Lite

In Chapter 3 we proved how simpler feed forward architectures achieve comparable performance to those using RNNs. We have also shown that the first CNNs based block of the architecture, trained with a small temporal context (90-seconds) single-channel EEG signal, does most of the work on a small-sized database. Hence, the

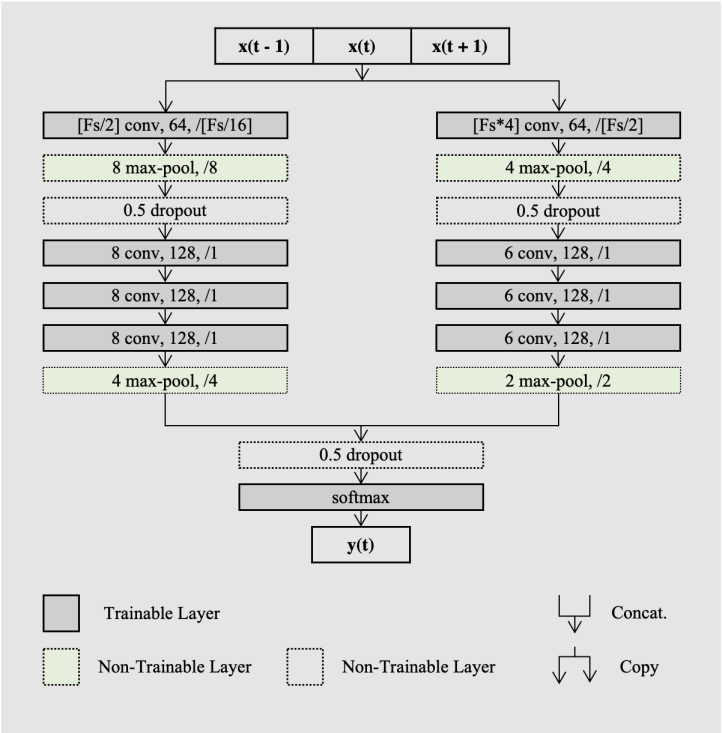


FIGURE 4.1: **DSN-L architecture.** An overview of the *representation learning* architecture from [29], with our *sequence-to-epoch* input-output training approach.

architecture we propose below - which we will refer as DSN-L - is the same as the simplest *3-SeqToEpoch-SPB* proposed in the previous chapter. The two architectures mainly differ in the optimization techniques used during the training procedure. In the following subsections we briefly describe the architecture, the training algorithm and the regularization techniques used in our scoring system.

4.1.1 Architecture

The architecture consists of two parallel *CNNs* employing small (CNN_{θ_S}) and large (CNN_{θ_L}) filters at the first layer. The small filter has been used to extract high-time resolution patterns, while the large filter has been used to extract high-frequency resolution patterns. The idea behind the use of the small and large filter sizes comes from the way the signal processing experts define the trade-off between temporal and frequency precision in the feature extraction procedure [85]. Each CNN section consists of four convolutional layers and two max-pooling layers. Each convolutional layer executes three operations: a one-dimensional convolution of the filters with the 90-second EEG signal, a batch normalization [86] and an element-wise rectified linear unit (ReLU) activation function. The pooling layer is used to down-sample the input. The filters size, the number of filters, the stride size of each *conv* layer, the pooling size and the stride size of the pooling layers are all defined in Figure 4.1.

The 90-second single-channel EEG signal $\mathbf{x}_i$ is given in input to the convolutional

neural networks CNN_{θ_S} and CNN_{θ_L}. The parameters θ of each CNN are independently trained, so as to return in output two feature vectors $\mathbf{h_i}^S$ and $\mathbf{h_i}^L$. The outputs are concatenated in $\mathbf{f_i}$, then forwarded to the *softmax* layer.

$$\mathbf{h_i}^S = CNN_{\theta_S}(\mathbf{x_i}) \tag{4.1}$$

$$\mathbf{h_i}^L = CNN_{\theta_L}(\mathbf{x_i}) \tag{4.2}$$

$$\mathbf{f_i} = \mathbf{h_i}^S || \mathbf{h_i}^L \tag{4.3}$$

The softmax function, together with the cross-entropy loss function, is used to train the model to output the logits $\mathbf{z_i}$ and the probability for the five mutually exclusive classes that correspond to the five sleep stages.

$$\mathbf{z_i} = \mathbf{W}^T \mathbf{f_i} + \mathbf{b} \tag{4.4}$$

$$\hat{p}_{i,k} = \frac{exp(z_{i,k})}{\Sigma_j exp(z_{i,j})} \tag{4.5}$$

where $\theta = \{\boldsymbol{W},\boldsymbol{b}\}$ are the parameters of the softmax layer, j is the index of the vector $\mathbf{z}$, $\hat{p}_{i,k}$ is the output probability of class k associated to $x(t)$, the centred 30-second signal in $\mathbf{x_i}$.

All the model specifications are reported in Figure 4.1, equally to the first *representation learning* in [29].

The architecture is trained end-to-end via backpropagation, using the sequence-to-epoch learning approach. The classification algorithms learn to predict the most represented class in the training set, leading to the so called class imbalance problem. Here the least represented classes are balanced by using two techniques: *(i) data augmentation*, by flipping vertically the data input (*i.e.*, multiply by -1 the original signal, see Figure 4.2) belonging to the least represented classes, then *(ii) oversampling* randomly the data so that all the sleep stages are equal in number to the most represented class. In our model, the input is a sequence of three 30-second epochs, and the output is the corresponding target of the central epoch at time t. So, we refer to the target of the central epoch to compute the most or least represented classes. The model is trained using mini-batch Adam gradient-based optimizer [91] with a learning rate lr. The training procedure runs up to a maximum number of iterations, as long as the break early stopping condition is satisfied - further details in the next subsection 4.1.2.

4.1.2 Regularization Techniques

Dropout. Commonly used as regularizer in CNNs, it prevents overfitting and co-adaptation of the feature maps [92]. During the training procedure a certain number of neurons are randomly removed, dropping units with a probability p. We fix the probability of dropping a connection equal to 50%, *i.e.*, $p = 0.5$.

Early stopping. It provides guidance on how many iterations can be run before the model begins to overfit [93]. The training procedure should be stopped as soon as the performance (*i.e.*, F1-score) on the validation set is lower than it was in the previous iteration step. However, in our experiments, before hastily stopping the learning procedure, the algorithm runs for an additional number of iterations (by

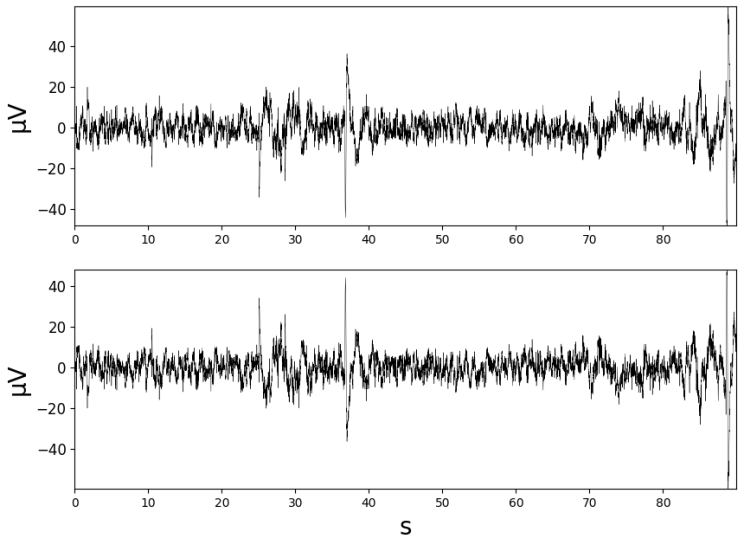


FIGURE 4.2: **Data augmentation by EEG vertical flip.** *(top)* 90-second EEG raw signal. *(bottom)* 90-second EEG vertically flipped.

fixing the so called *patience* parameter). The model with the highest performance is the one we finally save.

L2 weight decay. This technique simply adds a term to the loss function that penalizes the weight values; by doing so it avoids the exploding gradient phenomena [94]. The *lambda* defines the degree of penalty and it has been set to 10^{-3}.

4.1.3 Training Parameters

The training parameters are fixed as in [29]. The Adam optimizer's parameters *beta*1 and *beta*2 have been set to 0.9 and 0.999 respectively. The mini-batch size has been set to 100. During the batch normalization procedure, the ϵ value of 10^{-5} has been added to the mini-batch variance. In order to compute the mean and variance of the training samples, the moving average has been implemented using a fixed decay rate value of 0.999. The learning rates parameter lr has been fixed to 10^{-4}. The maximum number of iterations has been set to 100, with the early stopping *patience* parameter equal to 50.

The architecture has several hyperparameters (*e.g.*, number of layers, number/sizes of filters, regularization parameters, training parameters etc.) which could be optimized to tune its performance on any dataset. We decided not to systematically tune all these parameters - out of our scope - but to fix them for all the experiments, as done in the original networks.

4.2 Model Calibration

Along with the estimated sleep stage, the model should also provide a calibrated confidence - *i.e.*, the probability associated to the predicted stage should mirror its

ground truth correctness likelihood. We adopted label smoothing [95] to calibrate our model. It has been shown to be a suitable technique to improve model calibration [96].

In a standard training of a neural network, the cross-entropy loss is minimized using the hard targets y_k (*i.e.*, hot encoded targets, '1' for the correct class and '0' for the other). For a network trained with label smoothing, the hard targets are weighted with the uniform distribution $1/K$ (eq. 4.6), and the cross-entropy loss is minimized using the weighted mixture of the targets (eq. 4.7).

$$y_k{}^{LS_U} = y_k \cdot (1-\alpha) + \alpha/K \tag{4.6}$$

$$H(\mathbf{y}, \mathbf{p}) = \sum_{k=1}^{K} -y_k{}^{LS_U} \cdot \log(\hat{p}_k) \tag{4.7}$$

where α is the smoothing parameter, K is the total number of classes, $y_k{}^{LS_U}$ the targets smoothed with the uniform distribution, and $\hat{p}_k$ the softmax output probabilities.

In our experiments, we introduce a new distribution to smooth the labels, by mainly taking into account the importance in sleep scoring of the transitions from one sleep stage to the other. The idea is to compute the *conditional probability distribution* over the five sleep stages of all the sequences of epochs:

$$\boldsymbol{M} = P(stage(t)|stage(t-1), stage(t+1)) \tag{4.8}$$

where in $\boldsymbol{M}$ we have the conditional probability values for each possible combination of sequences of three sleep stages. In detail, we compute the probability to be in a stage at time t given the previous $(t-1)$ and the next $(t+1)$ sleep stages over the whole database. The matrix $\boldsymbol{M}$ is $(K \times K \times K)$ dimensional, where K is the total number of sleep stages.

As stated previously, the architecture takes in input a sequence of three epochs, and outputs the corresponding target of the central epoch $y_{k,(t)}$. So, during the training procedure, given the knowledge of the sleep stage at time $(t-1)$ and the sleep stage at time $(t+1)$, the hot encoded $y_{k,(t)}$ will be smoothed with the corresponding conditional probability vector from $\mathbf{M}$.

In Table 4.1 we report an example of the conditional probability values computed over the sequences extracted from the SEDF-SC-13 dataset (see section 4.4), with the label at time $(t-1)$ fixed in sleep stage awake. We highlight in light-green an example of the conditional probability vector to use when we had awake *W* at time $(t-1)$ and *N1* at time $(t+1)$, which results in the following smoothed target:

$$y_k{}^{LS_S} = y_k \cdot (1-\alpha) + \alpha \cdot \boldsymbol{M}_{W,K,N1} \tag{4.9}$$

The cross-entropy loss is minimized using the weighted mixture of the hard targets with these conditional probability distributions.

The smoothing parameter α for the uniform distribution and the *conditional probability distribution* weighting has been set to 0.1 and 0.2 respectively. These two values

TABLE 4.1: **Conditional probability values on sleep sequences.** Conditional probability values computed over the sequences, extracted from the SEDF-SC-13 ±30mins dataset, with the label at time $(t-1)$ fixed in awake. i.e., $\boldsymbol{M}_{W,K\times K}$.

W(t-1)	*W(t+1)*	*N1(t+1)*	*N2(t+1)*	*N3(t+1)*	*R(t+1)*
W(t)	0.991	0.503	0.131	0.333	0.217
N1(t)	0.008	0.495	0.581	0.000	0.109
N2(t)	0.000	0.002	0.275	0.000	0.000
N3(t)	0.000	0.000	0.006	0.667	0.000
R(t)	0.000	0.000	0.006	0.000	0.674

gave us the highest performance on both SEDF-SC-13 and SEDF-SC-18. In both, we explored α values up to 0.5.

4.3 Monte Carlo Dropout

We exploit the dropout regularization technique to both enhance the performance of the model and to estimate the model uncertainty. As explained above, during the training procedure, at each iteration, dropout removes a certain number of units within our network at random. It randomly samples a certain number of sub-networks, so that each time the model's architecture is slightly different. In a standard application, dropout is used only during the training phase. At test time, instead, all the trained neurons and connections are used - *i.e.*, all the weights of the whole network. The output could be interpreted as an averaging ensemble of all the sub-networks.

We employ, for the first time in sleep staging, the *Monte Carlo* (*MC*) dropout [97], to quantify the model uncertainty, and to further enhance the performance of the scoring architecture. Monte Carlo refers to a specific class of algorithms that rely on random sampling, to provide estimates and distributions of numerical quantities. *MC dropout* simply consists in applying the randomized sampling even at test time. The different sub-networks could be interpreted as *Monte Carlo* samples extracted from the space of all the possible models. As a result, by applying dropout N times at inference time (with the probability of dropping a connection $p = 0.5$), we would get N different predictions. We compute the mean and the variance of the N predictions for each sleep stage k using the following

$$\mu_{\mathrm{i,k}} = \frac{\sum_{n=1}^{N} \hat{p}_{\mathrm{n,i,k}}}{N} \tag{4.10}$$

$$\sigma^2{}_{\mathrm{i,k}} = \frac{\sum_{n=1}^{N} (\hat{p}_{\mathrm{n,i,k}} - \mu_{\mathrm{i,k}})^2}{N} \tag{4.11}$$

where $\hat{p}_{\mathrm{n,i,k}}$ is the output probability for the sleep stage k of the n-th prediction for the input $\mathbf{x}_{\mathrm{i}}$. The final prediction $\hat{y}_{\mathrm{i}}$ of the model will be given by $max(\boldsymbol{\mu}_{\mathrm{i}})$, which we will refer to as μ_{max}, along with the assigned variance value $\sigma^2{}_{\mu\mathrm{max}}$.

TABLE 4.2: **Sleep stages on SEDF-SC-13 and SEDF-SC-18.**
Number and percentage of 30-second epochs per sleep stage of the SEDF-SC datasets with different trimming.

Datasets	W	N1	N2	N3	R	Total
SEDF-SC-13 ±30mins	8285 (19.6%)	2804 (6.6%)	17799 (42.1%)	5703 (13.5%)	7717 (18.2%)	42308
SEDF-SC-13	5907 (15.2%)	2687 (6.9%)	17255 (44.3%)	5465 (14.0%)	7647 (19.6%)	38961
SEDF-SC-18 ±30mins	65951 (33.7%)	21522 (11.0%)	69132 (35.4%)	13039 (6.7%)	25835 (13.2%)	195479
SEDF-SC-18	43055 (26.3%)	19168 (11.7%)	64408 (39.3%)	12042 (7.3%)	25275 (15.4%)	163948

The uncertain predictions will be then estimated by analysing both their computed mean and variance. The selection procedure of the uncertain sleep stages is explained in detail in subsection 4.5.4. The selected uncertain predictions could be then presented to the physician for a secondary review.

4.4 Database

SEDF-SC. The Sleep-EDF Sleep Cassette is a subset of the open source Sleep-EDF dataset [32], [48]. The PSG data belong to 78 subjects (37 males and 41 females) aged from 25 to 101 years. Except for the first nights of subjects 36 and 52, and for the second night of subject 13, for all the subjects are available two whole nights, resulting in 153 PSG recordings. Each recording includes two scalp EEG channels (Fpz-Cz and Pz-Cz), one EOG (horizontal) channel, one submental chin EMG channel and one oro-nasal respiration channel. The recordings are manually scored by sleep experts on 30-second epochs according to R&K scoring rules [9], resulting in the eight classes Wake, N1, N2, N3, N4, REM, MOVEMENT and UNKNOWN. In order to use the AASM standard [2], we have merged the N3 and N4 stages into a single stage N3, and we have excluded the MOVEMENT and UNKNOWN classes. In many recordings there were long wake periods before the patients went to sleep and after they woke up. We have done experiments with the two common ways these periods are trimmed in literature: 1) only *in-bed* parts are employed [98], *i.e.*, from *light-off* time to *light-on* time; 2) 30 minutes of data before and after *in-bed* parts are taken into account in the experiments [29]. In our experiments we have considered the single-channel EEG Fpz-Cz, with a sampling rate of 100 Hz and without any pre-processing.

In order to facilitate the comparison with many existing DL based scoring algorithms, in this work we use the last expanded version published in 2018 (to which we will refer as **SEDF-SC-18**), and also the previous upload of the Sleep-EDF database published in 2013 (to which we will refer as **SEDF-SC-13**). In the older upload there were only 39 PSG recordings from 20 subjects. In Table 4.2 we report a summary of the total number and percentage of the epochs per sleep stage.

TABLE 4.3: **Data split on SEDF-SC-13 and SEDF-SC-18.**
Data split on the SEDF-SC datasets.

Datasets	Size	Experimental Setup	Held-out Validation Set	Held-out Test Set
SEDF-SC-13	20	20-fold CV	4 subjects	1 subject
SEDF-SC-18	78	10-fold CV	7 subjects	7 subjects

4.5 Results

4.5.1 Experiment Designs

The validation procedure is in line with the state-of-the-art methods considered in Table 4.7. In fact, we evaluate our model using the k-fold cross-validation scheme. We set k equal to 20 for SEDF-SC-13 (leave-one-out evaluation procedure) and 10 for SEDF-SC-18 datasets. In Table 4.3 we summarize the data split for each dataset. We decide to further standardize the experiments by considering in each fold the same subject IDs used in [73]. We believe that in such small datasets, the subjects involved in the training/validation/test set may have an impact on the final results.

The following experiments are conducted:

- ***base***. The model is trained without label smoothing.
- ***base+LS*$_{\mathbf{U}}$**. The model is trained with label smoothing using the standard $1/K$ uniform distribution - *i.e.*, the hard targets are weighted with the uniform distribution.
- ***base+LS*$_{\mathbf{S}}$**. The model is trained with label smoothing using our statistical analysis done on the sequences of sleep stages - *i.e.*, the hard targets are weighted with the *conditional probability distribution*.

These three models, differently trained, have been evaluated with and without the *MC dropout* sampling technique. In subsection 4.5.3 we present the results obtained for the three models, and the impact of *MC dropout* at inference time.

4.5.2 Metrics

The per-class F1-score, the overall accuracy (*Acc.*), the macro-averaging F1-score (*MF1*) and the Cohen's kappa (k) have been computed from the predicted sleep stages from all the folds to evaluate the performance of our model [88], [89]. In our experiments the weighted-averaging F1-score has been also reported, taking into account also the label imbalance problem. It computes the average of the metric weighted by the number of true instances for each label. The F1-score computed in this way is not a realistic weighted average of the precision and recall, but it takes into account the high imbalance between the sleep stages.

Model Calibration. We evaluated the calibration of our model using the expected calibration error (ECE) proposed in [90]. It approximates the difference in expectation between accuracy acc and confidence $conf$, where with confidence it

TABLE 4.4: **DSN-L models performance.** Overall performance and calibration measure of the models obtained from 20-fold and 10-fold cross-validation with and without *MC* on both SEDF-SC-13 ±30mins and SEDF-SC-18 ±30mins datasets. Best shown in bold.

			Overall Metrics				Calibration	
Datasets		Models	Acc.	MF1	k	F1	ECE	*conf*
SEDF-SC-13 ±30mins	*w/o MC*	*base*	82.3%	76.6%	0.76	82.5%	0.111	93.4%
		base+LS_U	**82.8%**	**77.2%**	**0.77**	**83.0%**	**0.023**	80.5%
		base+LS_S	82.7%	76.4%	0.76	82.7%	0.071	89.7%
	w/ MC	*base*	83.0%	77.1%	0.77	83.0%	0.060	89.0%
		base+LS_U	**84.0%**	**78.0%**	**0.78**	**83.9%**	0.055	78.5%
		base+LS_S	83.4%	77.0%	0.77	83.3%	**0.031**	86.5%
SEDF-SC-18 ±30mins	*w/o MC*	*base*	**79.4%**	**74.5%**	**0.72**	**80.0%**	0.064	85.8%
		base+LS_U	79.3%	74.3%	**0.72**	79.8%	**0.020**	77.4%
		base+LS_S	79.2%	74.4%	**0.72**	79.8%	0.045	83.8%
	w/ MC	*base*	80.3%	**75.3%**	**0.73**	**80.7%**	0.031	83.4%
		base+LS_U	80.3%	75.2%	**0.73**	80.6%	0.047	75.6%
		base+LS_S	**80.4%**	**75.3%**	**0.73**	**80.7%**	**0.015**	81.9%

refers to the softmax output probabilities. More in detail, we first divide the predictions into M equally spaced bins (size $1/M$), then for each bin we compute the accuracy $acc(B_m)$ and we define the average predicted probability value $conf(B_m)$:

$$acc(B_m) = \frac{1}{|B_m|} \cdot \sum_{i \in B_m}^{K} \mathbf{1}(\hat{y}_i = y_i) \tag{4.12}$$

$$conf(B_m) = \frac{1}{|B_m|} \cdot \sum_{i \in B_m}^{K} \hat{p}_i \tag{4.13}$$

where y_i and $\hat{y}_i$ are the true and predicted labels for the sample i, B_m is the group of samples whose predicted probability values falls into the interval $I_m = (\frac{m-1}{M}, \frac{m}{M}]$, and $\hat{p}_i$ is the predicted probability value for sample i.
Then we finally compute the weighted average of the *acc* and *conf* difference of the M bins,

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} \cdot |acc(B_m) - conf(B_m)| \tag{4.14}$$

where n is the number of samples in each bin.

Clearly, perfectly calibrated models have $acc(B_m) = conf(B_m)$ for all $m \in \{1, .., M\}$, resulting in $ECE = 0$.

4.5.3 Analysis of Experiments

In Table 4.4 we report the overall performance and the calibration measure of three different models, with and without Monte Carlo dropout at inference time, to which

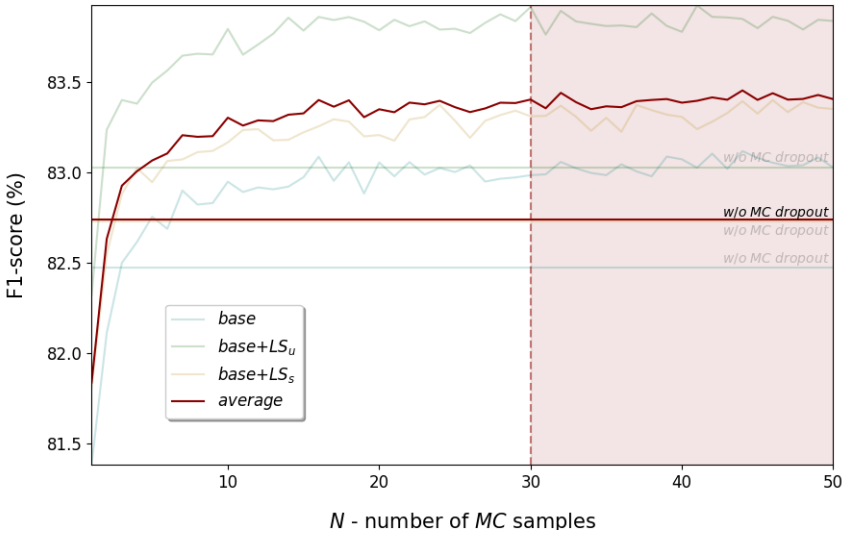


FIGURE 4.3: **F1-score and Monte Carlo Sampling.** F1-score against the number of *Monte Carlo* samples N of the three models (*base*, *base+LS*$_{\mathrm{U}}$ and *base+LS*$_{\mathrm{S}}$) evaluated on SEDF-SC-13 ±30mins dataset. *Monte Carlo* sampling converges after 30 samples without further significant improvement on the average of the three models.

we refer *w/ MC* and *w/o MC* respectively, on both SEDF-SC-13 ±30mins and SEDF-SC-18 ±30mins datasets. We show the efficiency of label smoothing in calibrating the model. The $conf$ value refers to the average of all the predicted probability values. In both LS_{U} and LS_{S} models, the $conf$ probability better reflects the ground truth correctness likelihood - *i.e.*, accuracy value. Indeed, *e.g.*, on the SEDF-SC-13 ±30mins dataset, it results in a better ECE value 0.023 and 0.071, compared to the higher 0.111 for the *base* model. The overall performance are preserved or even improved. By using *MC* at test time, we show the efficiency of label smoothing and *MC* techniques in both calibrating and enhancing the performance of the model. It is quite interesting the impact of *MC dropout*: an increase in overall metrics and a decrease in the average predicted probability values. This justifies a better calibrated model by using our *conditional probability distribution* smoothing technique LS_{S} - *i.e.*, on the SEDF-SC-13 ±30mins dataset, ECE value equal to 0.031. In Figure 4.3 we report the F1-score against the number of *Monte Carlo* samples N, evaluated over all the experiments experiments performed on the SEDF-SC-13 ±30mins dataset. We want to highlight how the *Monte Carlo* sampling outperforms the experiments done without applying *MC* after approximately three samples, on the average of the three models. On average we get a plateau after 30 samples, so we decided to set N equal to 30 in all our experiments *w/ MC*.

From here on, the analysis will be limited to one of the experimented models, *i.e.*, the *base+LS*$_{U}$ *w/ MC*, on average the model with high performance on both the SEDF-SC datasets.

In Table 4.5 we report the confusion matrix and the per-class performance of the best of our models evaluated on SEDF-SC-13 ±30mins and SEDF-SC-18 ±30mins respectively. The *i-th* row and the *j-th* column indicates the percentage number of

TABLE 4.5: **Confusion matrix on SEDF-SC-13 ±30mins and SEDF-SC-18 ±30mins.** Confusion matrix obtained from 20-fold and 10-fold cross-validation on both SEDF-SC-13 ±30mins and SEDF-SC-18 ±30mins datasets.

			Predicted				Per-class Metrics		
Dataset	(%)	W	N1	N2	N3	R	Pr.	Rec.	F1
SEDF-SC-13 ±30mins	W	**90.4**	6.0	1.4	0.3	1.9	84.0	90.4	87.1
	N1	18.6	**42.6**	19.2	0.8	18.8	46.5	42.6	44.4
	N2	3.0	2.4	**86.0**	4.5	4.1	89.9	86.0	87.9
	N3	1.3	0.2	7.9	**90.6**	0	85.9	90.6	88.2
	R	3.5	5.7	7.8	0.1	**82.9**	82.0	82.8	82.4

			Predicted				Per-class Metrics		
Dataset	(%)	W	N1	N2	N3	R	Pr.	Rec.	F1
SEDF-SC-18 ±30mins	W	**90.0**	7.5	0.6	0.2	1.7	93.0	90.0	91.5
	N1	14.2	**48.1**	24.5	1.0	12.2	44.1	48.1	46.0
	N2	0.7	8.5	**80.3**	5.3	5.2	85.6	80.3	82.9
	N3	0.2	0.3	12.8	**86.5**	0.2	73.0	86.5	79.2
	R	3.4	8.8	7.6	0.8	**79.4**	73.7	79.4	76.4

90-second EEG instances with the true label being *i-th* class and the predicted label being *j-th* class. In bold we highlight the percentage number of instances well classified. As expected [3], the lowest performance has been obtained for the N1 sleep stage, *i.e.*, F1-score 44.4% and 46.0%; most of the N1 have been wrongly classified in awake, N2 and REM. The F1-score for all the other sleep stages were in range between 82.4% and 88.2% on SEDF-SC-13 ±30mins, and between 76.4% and 91.5% on SEDF-SC-18 ±30mins.

4.5.4 Uncertainty estimate

In order to select the uncertain instances, at first, we used the variance, *i.e.*, ${\sigma^2}_{\mu\max}$ of the predicted probability values obtained from the N sampling. The selection procedure - referred to as query procedure - simply rely on the setting of a threshold value q%, that corresponds to the percentage number of epochs - for each PSG recording - to select (reject) and to send potentially to the physician for a secondary review. The epochs with the highest values of variance will be the q% selected. We also tried to use the mean, *i.e.*, $\mu_{\max}$ of the predicted probability values obtained from the N sampling, to select the uncertain instances. In this case the epochs with the lowest mean values will be the q% selected. The selected epochs are the predictions where the averaging ensemble of the models outputs the higher uncertainty.

In Figure 4.4 we report the F1-score computed over the remaining epochs against the percentage number of selected instances. We have fixed the q% threshold value to 5%, because it was considered to be a reasonable number of epochs (54 on average for each PSG recording) to select and to eventually present to the physician for a secondary review. The results show that by using $\mu_{\max}$ in the selection procedure we obtain higher performance. In Figure 4.5 we report, for each q% number of selected instances, the percentage of misclassified and correctly classified epochs among the selected ones. As illustrated, by using μ, the percentage number of misclassified

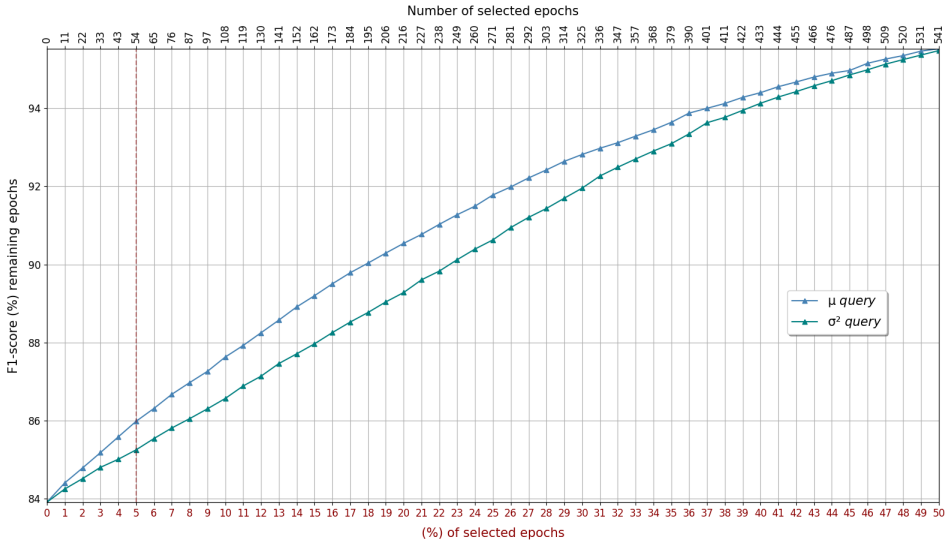


FIGURE 4.4: **F1-score after query procedure.** F1-score computed over the remaining epochs after the query procedure against the percentage number of epochs to select. In light green and in light blue the F1-score performance in case the selection procedure has been done using the variance ($\sigma^2{}_{\mu\max}$ query) and the mean ($\mu_{\max}$ query) respectively. The performance refers to the best of our model evaluated on SEDF-SC-13 ±30mins dataset.

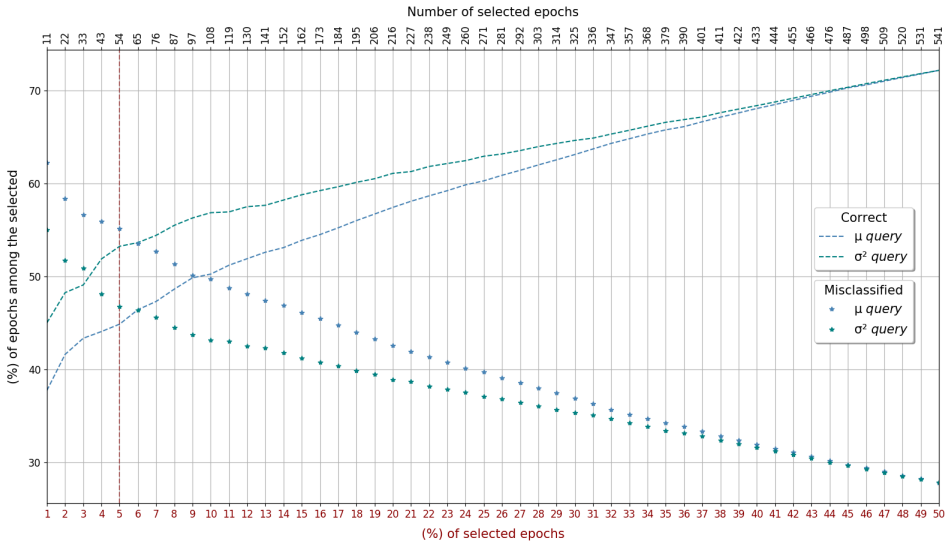


FIGURE 4.5: **Percentage of selected/rejected correct/misclassified epochs.** Percentage of misclassified and correctly classified epochs among the q% selected. In light green and in light blue the percentage values in case the selection procedure has been done using the variance ($\sigma^2{}_{\mu\max}$ query) and the mean ($\mu_{\max}$ query) respectively. The performance refers to the best of our models evaluated on SEDF-SC-13 ±30mins dataset.

TABLE 4.6: **Per-class $\sigma^2{}_{\mu\text{max}}$ and μ_{max} on prediction w/ *MC*.**
Per-class $\sigma^2{}_{\mu\text{max}}$ and μ_{max} of the predicted probability values computed on SEDF-SC-13 ±30mins and SEDF-SC-18 ±30mins datasets.

Datasets	Total	W	N1	N2	N3	R
SEDF-SC-13	$\sigma^2{}_{\mu\text{max}}$	0.008	0.024	0.009	0.006	0.014
±30mins	μ_{max}	81.3%	60.2%	80.4%	81.7%	75.3%
SEDF-SC-18	$\sigma^2{}_{\mu\text{max}}$	0.005	0.015	0.009	0.006	0.013
±30mins	μ_{max}	82.9%	60.4%	74.7%	79.5%	70.4%

epochs are greater than the correctly classified up to the selection threshold q% equal to 10%. Whilst, by using $\sigma^2{}_{\mu\text{max}}$, the percentage number of selected epochs q% radically decreases to 2%.

In Table 4.6 we also report the average of the per-class $\sigma^2{}_{\mu\text{max}}$ and μ_{max} predicted probability values, to have an overall estimate of the model uncertainty, evaluated on both SEDF-SC-13 ±30mins and SEDF-SC-18 ±30mins datasets. As expected, the results show that the model has more difficulty in classifying N1 and REM epochs, while provides greater confidence in classifying W, N2 and N3 sleep stages (lower variance and higher predicted probability values).

4.5.5 Benchmarking with SOTA

In Table 4.7 we compare our best model with the other state-of-the-art methods evaluated on the two versions of the SEDF-SC database. We report the results for each experimental scenario: 1) only *in-bed* recordings; 2) additional 30 minutes recordings before and after *in-bed*. We have considered only the methods using DL based architectures, raw single-channel EEG Fpz-Cz, same evaluation procedure (*i.e.*, k-fold cross-validation) and using independent training and test sets. We decided to further standardize our experiments by considering in each fold the same subject IDs used in [73]. All the results indicated by † are not directly comparable, since they use a different set of subject IDs in their training/ evaluation/ testing procedure. The sleep scoring algorithms are compared across the overall metrics (Acc., MF1, Cohen's Kappa and F1-score) and the per-class F1-score. The proposed DSN-L achieves slightly lower performance, if not on par, compared to the state-of-the-art models on all the SEDF-SC datasets. The results confirm what we had already partially observed in [7] on the SEDF-SC-13: the first *EPB* block from DeepSleepNet, trained with a small temporal context in input, still succeed in solving the classification task on the small-sized database. Indeed, on both SEDF-SC-13 and SEDF-SC-18 *in-bed* recordings, our model achieves an overall accuracy only below 1.3% compared to the recent state-of-the-art XSleepNet2 [73]. We are not surprised to see our lighter architecture to overperform DeepSleepNet: one of the reasons could be that in [29] they have not implemented any early stopping procedure, and they save their model only at the latest iteration step, thus not mitigating the overfitting phenomenon. The number of training parameters of our lighter model are significantly reduced, ∼0.6M compared to the others TinySleepNet [72] ∼1.3M, SleepEEGNet [69] ∼2.6M, FCNN+RNN [73] ∼5.6M, Naive Fusion and XSleepNet2 ∼5.8M [73] and DeepSleepNet [29] ∼24.7M. Nevertheless, SeqSleepNet [71] is still the network with

TABLE 4.7: **Benchmarking DSN-L with SOTA.** Comparison between our method and the other DL based automatic sleep scoring systems using raw single-channel EEG Fpz-Cz, evaluated on SEDF-SC datasets with overall accuracy (%Acc.), macro F1-score (%MF1), Cohen's Kappa (k) and % per-class F1-score. The best performance metrics for each dataset are indicated in bold.

Datasets	Methods	Training Param.	Sequences of epochs	Overall Metrics			Per-class F1-score				
				Acc.	MF1	k	W	N1	N2	N3	R
SEDF-SC-13 ±30mins	FCNN+RNN [73]	∼ 5.6M	20	81.8	75.6	0.75	89.4	44.1	84.0	84.0	76.3
	DeepSleepNet [29]	∼ 24.7M	25	81.4	75.6	0.75	83.9	43.5	85.4	84.1	81.1
	DeepSleepNet [29] †	∼ 24.7M	25	82.0	76.9	0.76	84.7	46.6	85.9	84.8	82.4
	IITNet [68] †	-	10	84.0	77.7	0.78	87.9	44.7	88.0	85.7	82.1
	Our method	∼ 0.6M	**3**	84.0	78.0	0.78	87.1	44.4	87.9	88.2	82.4
	SleepEEGNet [69] †	∼ 2.6M	10	84.3	79.7	0.79	89.2	**52.2**	86.8	85.1	**85.0**
	SeqSleepNet+ [71]	∼ **0.2M**	20	85.2	78.4	0.80	90.5	45.4	**88.1**	86.4	81.8
	Naive Fusion [73]	∼ 5.8M	20	85.0	78.8	0.79	91.7	48.8	87.2	82.9	83.6
	TinySleepNet [72] †	∼ 1.3M	15	85.4	80.5	0.80	90.1	51.4	88.5	**88.3**	84.3
	XSleepNet2 [73]	∼ 5.8M	20	**86.3**	**80.6**	**0.81**	**92.2**	51.8	88.0	86.8	83.9
SEDF-SC-13	Naive Fusion [73]	∼ 5.8M	20	80.2	74.9	0.72	77.3	47.4	85.8	84.8	79.3
	DeepSleepNet [29]	∼ 24.7M	25	82.5	76.8	0.76	80.1	47.3	87.0	85.7	83.8
	DeepSleepNet [29] †	∼ 24.7M	25	82.6	77.1	0.76	**82.9**	46.8	86.5	84.1	85.2
	FCNN+RNN [73]	∼ 5.6M	20	81.3	76.0	0.74	76.4	50.0	86.8	85.3	81.3
	SeqSleepNet+ [71]	∼ **0.2M**	20	82.2	74.1	0.75	78.5	37.1	87.6	86.2	81.2
	Our method	∼ 0.6M	**3**	82.6	76.3	0.76	81.6	42.4	87.4	**87.9**	82.1
	XSleepNet2 [73]	∼ 5.8M	20	**83.9**	**78.7**	**0.77**	81.6	**52.9**	**88.1**	85.3	**85.4**
SEDF-SC-18 ±30mins	DeepSleepNet [29]	∼ 24.7M	25	76.9	70.7	0.69	90.8	44.8	78.5	67.9	71.3
	DeepSleepNet [29] †	∼ 24.7M	25	77.1	71.2	0.69	90.4	46.0	79.1	68.6	71.8
	SleepEEGNet [69] †	∼ 2.6M	10	80.0	73.6	0.73	91.7	44.1	82.5	73.5	76.1
	Our method	∼ 0.6M	**3**	80.3	75.2	0.73	91.5	46.0	82.9	79.2	76.4
	Naive Fusion [73]	∼ 5.8M	20	82.3	76.2	0.75	93.2	49.6	**86.2**	79.4	**82.5**
	SeqSleepNet+ [71]	∼ **0.2M**	20	82.6	76.4	0.76	92.2	47.8	84.9	77.2	79.9
	FCNN+RNN [73]	∼ 5.6M	20	82.8	76.6	0.76	92.5	47.3	85.0	79.2	78.9
	TinySleepNet [72] †	∼ 1.3M	15	83.1	**78.1**	0.77	92.8	**51.0**	85.3	**81.1**	80.3
	XSleepNet2 [73]	∼ 5.8M	20	**84.0**	77.9	**0.78**	**93.3**	49.9	86.0	78.7	81.8
SEDF-SC-18	DeepSleepNet [29]	∼ 24.7M	25	76.0	72.2	0.68	88.1	45.8	79.7	74.3	72.9
	DeepSleepNet [29] †	∼ 24.7M	25	76.6	73.0	0.69	88.3	46.1	79.9	76.2	74.4
	SeqSleepNet+ [71]	∼ **0.2M**	20	79.0	74.6	0.71	83.2	46.8	85.5	76.3	81.0
	Our method	∼ 0.6M	**3**	79.0	75.1	0.72	**89.3**	46.9	83.3	78.9	77.1
	Naive Fusion [73]	∼ 5.8M	20	79.1	75.1	0.71	83.9	47.8	85.4	78.4	79.8
	FCNN+RNN [73]	∼ 5.6M	20	79.3	75.1	0.71	84.2	49.1	85.2	76.8	80.4
	XSleepNet2 [73]	∼ 5.8M	20	**80.3**	**76.4**	**0.73**	85.2	**49.4**	**86.0**	**79.8**	**81.7**

the lowest number of parameters ∼0.2M. We did not report the number of training parameters for IITNet [68] since it was not available in literature.

4.5.6 Benchmarking among our methods

In Table 4.8 we report the results of our best model evaluated on the two versions of the SEDF-SC database - in both experimental scenarios. The outcomes refer to the performance of the model evaluated before the selection procedure and after the selection procedure, by using $\sigma^2{}_{\mu\max}$ and $\mu_{\max}$ query values. We report the results obtained after the selection procedure on both the kept and rejected set of epochs. As a consequence of what we have observed in Figure 4.5, on both SEDF-SC-13 and SEDF-SC-18, the model shows an increase in performance over the kept epochs, and a significant decrease on the rejected epochs (below 50% by using $\mu_{\max}$ query). These results highlight the efficiency of the query procedure to select a larger number of misclassified epochs among the selected one. The best performance for each dataset

TABLE 4.8: **Benchmarking DSN-L after query procedure w/ *MC*.** Comparison among our methods using ${\sigma^2}_{\mu\max}$ and $\mu_{\max}$ query selection procedures (q% threshold value fixed to 5%), evaluated on SEDF-SC datasets with overall accuracy (%Acc.), macro F1-score (%MF1), Cohen's Kappa (k), weighted-averaging F1-score (%F1) and % per-class F1-score. The best performance metrics for each dataset are indicated in bold.

Datasets		Evaluated Epochs	Overall Metrics				Per-Class F1-Score				
			Acc.	MF1	k	F1	W	N1	N2	N3	R
SEDF-SC-13 ±30mins	-	all	84.0	78.0	0.78	83.9	87.1	44.4	87.9	88.2	82.4
	${\sigma^2}_{\mu\max}$	kept	85.7	77.9	0.80	85.2	88.6	39.6	88.9	88.5	84.2
		rejected	53.0	47.5	0.36	52.4	38.7	54.9	53.3	32.3	58.1
	$\mu_{\max}$	kept	**86.1**	**79.6**	**0.81**	**86.0**	**89.1**	**45.7**	**89.4**	**89.0**	**84.8**
		rejected	44.7	43.4	0.28	44.8	42.8	39.3	49.2	37.9	47.8
SEDF-SC-13	-	all	82.6	76.3	0.76	82.4	81.6	42.4	87.4	87.9	82.1
	${\sigma^2}_{\mu\max}$	kept	84.3	76.4	0.78	83.8	83.5	37.9	88.4	88.3	83.6
		rejected	50.0	45.2	0.32	49.4	39.6	51.9	47.2	30.6	56.8
	$\mu_{\max}$	kept	**84.5**	**77.8**	**0.78**	**84.4**	**83.6**	**43.5**	**88.8**	**88.7**	**84.4**
		rejected	45.6	45.7	0.29	45.8	45.9	37.6	52.2	49.3	43.5
SEDF-SC-18 ±30mins	-	all	80.3	75.2	0.73	80.6	91.5	46.0	82.9	79.2	76.4
	${\sigma^2}_{\mu\max}$	kept	81.7	75.9	0.75	81.8	92.3	45.6	84.0	79.9	77.5
		rejected	55.2	51.0	0.40	54.4	49.9	49.0	48.1	39.5	68.6
	$\mu_{\max}$	kept	**82.3**	**76.7**	**0.76**	**82.5**	**92.6**	**47.1**	**84.4**	**80.1**	**79.4**
		rejected	42.8	41.8	0.24	43.0	44.3	39.4	45.8	35.6	43.8
SEDF-SC-18	-	all	79.0	75.1	0.72	79.3	89.3	46.9	83.3	78.9	77.1
	${\sigma^2}_{\mu\max}$	kept	80.2	75.8	0.73	80.4	90.0	46.9	84.3	79.8	77.8
		rejected	57.1	52.1	0.42	56.4	48.9	47.7	50.7	41.4	71.7
	$\mu_{\max}$	kept	**80.9**	**76.7**	**0.74**	**81.2**	**90.7**	**48.0**	**84.7**	**79.8**	**79.7**
		rejected	42.4	41.1	0.23	42.6	41.3	39.7	44.9	34.6	44.9

are indicated in bold. We obtain an overall accuracy equal to 86.1% on SEDF-SC-13 ±30mins (84.5% on *in-bed* only) and equal to 82.3% on SEDF-SC-18 ±30mins 80.9% on *in-bed* only).

4.6 Discussion

DSN-L achieves performance slightly lower, if not on par, compared to the existing state-of-the-art methodologies evaluated on the SEDF-SC databases.
Beside being trained on a small number of parameters, our method does not require any extra resources to buffer the sequences in input, since it processes sequences of only 90-seconds EEG. Therefore, we may assume that an automatic sleep scoring system does not necessarily have to encode such long temporal structures, rather intrinsic patterns of short-term PSG recordings may be sufficient. The Monte Carlo dropout technique allows us to enhance the performance of the architecture, and, at the same time, to identify a relevant number of misclassified epochs among the ones selected during the query procedure. The major advantage of the proposed approach is that it simultaneously enhance the performance of the architecture and it provides an estimate of the model uncertainty by exploiting existing layers of the

architecture. Unlike the existing confidence estimation algorithms for sleep scoring [52], [67], the proposed uncertainty estimate procedure is easy to implement and it does not require any additional computation over the baseline architecture. Moreover, it produces interpretable outputs, *i.e.*, mean and variance of the predicted probability values. Whilst, a clear disadvantage of the Monte Carlo dropout approach - as for other ensemble learning based algorithms - is that it needs to be executed N times, obviously increasing the evaluation time by N. However, in a real-time application, it may still be a valid solution because the evaluation of a single sequence takes only a few milliseconds.

The idea to train the model by smoothing the labels through the *conditional probability distribution* is interesting, however the results obtained in subsection 4.5.3 Table 4.4 should be further investigated. How this prior knowledge is affecting the learning procedure is unclear. Even if with this technique we succeed to better calibrate our network, we do not equally succeed in obtaining higher performance using it in combination with our query procedure. In addition, unlike what we expected, it is not always the case that a better calibrated architecture leads to a better estimate of the model uncertainty. A model with a lower ECE value (see Table 4.4) does not always enable the detection of a higher number of percentage of misclassified epochs (see Supplementary Tables B.2 and B.3). This last statement will be further investigated in the next Chapter 5.

As a result of follow-up analysis, we realized that the proposed uncertainty estimate is equally efficient in identifying the misclassified epochs, even with a baseline/standard query selection procedure (*i.e.*, *w/o MC*, the predicted probability values $max(\hat{\boldsymbol{p}}_i)$ are used to select the uncertain instances, see the high values - $> 50\%$ in percentage of misclassified epochs detected in Table B.2).

Although the proposed simple feed forward architecture has proven to be as efficient as RNNs based architectures, we cannot generalize concluding that by using only this first representation learning block we will reach equally good results on larger databases. As a result of further experiments carried out on larger and more heterogeneous databases (*e.g.*, PHYS [48], [49] and SHHS [34], [36]), we can state that these observations are mainly valid on small-sized dataset (*i.e.*, low heterogeneity between subjects). DSN-L has a low capacity, *i.e.*, low number of training parameters, hence is less prone to overfitting on a small dataset. Therefore the need to further investigate its robustness on larger database. It would be also interesting to simulate the query procedure on the recent state-of-the-art architectures to assess its benefit on them. In Chapter 6 we will test the proposed uncertainty estimate procedure, exploiting the model ensembling and the label smoothing techniques on a powerful and recently proposed state-of-the-art architecture.

Chapter 5

Exploitation of the multi-scored databases in automated sleep scoring

Most of the existing automated sleep scoring systems are trained using labels annotated by a single scorer, whose subjective evaluation is transferred to the model. The first remarkable exception comes from [67], where they consider recordings scored by six different physicians [99]. The scoring algorithm was trained on the six-scorer consensus (*i.e.*, based on the majority vote weighted by the degree of consensus from each physician, see section 5.2). In [25] the Dreem group introduced two publicly-available datasets scored by five sleep physicians. Similarly, they used the scorer consensus to train their automated scoring system. It has been shown that the performance of an automated sleep scoring system is on-par with the scorer consensus [25], [67], and mainly that their best scoring algorithm is better than the best human scorer - *i.e.*, the scorer with the higher consensus among all the physicians in the group.

Although they both considered the knowledge from the multiple scorers - by averaging their labels and by training their algorithm on the averaged consensus - they still trained the algorithm on a single one-hot encoded label. Indirectly, they are still transferring the best scorer's subjectivity into the model, and they are not explicitly training the model to adapt to the consensus of the group of scorers.

In Chapter 5 we train our DeepSleepNet-Lite (DSN-L) architecture and the lightweight SimpleSleepNet (SSN) architecture proposed in [25] on three open-access multi-scored sleep datasets. We consider the multiple-labels in the training procedure, *i.e.*, the annotations of all the physicians are taken into account at the same time. First we assess the performance of both scoring systems trained with the scorer consensus, and compare it to the performance of the individual scorer-experts. Then we exploit label smoothing along with the soft-consensus distribution to insert the multiple-knowledge into the training procedure of the models and to better calibrate the scoring architectures. We quantify the similarity between the hypnodensity-graph generated by the models - trained with and without label smoothing - and the hypnodensity-graph generated by the scorer consensus. We finally further analyze the ability of the uncertainty estimate and query procedure, proposed in Chapter 4, to identify the most challenging sleep stage predictions on our calibrated DSN-L, on both the model trained with and without smoothing their labels, whilst using a soft-consensus distribution. We aim at exploring if with a better calibrated model (*i.e.*, the predicted probability value $\hat{p}$ mirrors its ground truth

correctness likelihood) we are able to detect a higher number of misclassified epochs.

Contributions. Our contributions can be summarized as follows: (1) we demonstrate the efficiency of label smoothing along with the soft-consensus distribution in both calibrating and enhancing the performance of both DSN-L and SSN; (2) we show how the model can better resemble the scorer group consensus, leading to a similarity increase between the hypnodensity-graph generated by the model and the hypnodensity-graph generated by the scorer consensus; (3) we prove the efficiency of the query procedure in detecting the most challenging sleep epochs, and we found that with a better calibrated model we are not always able to better detect the misclassified epochs.

5.1 Architectures

We run our experiments on both DSN-L architecture (see subsection 4.1.1) and SSN architecture [25].

DSN-L has been already described in detail in the previous chapter. Below, we briefly describe the SSN architecture. For further details we refer the reader to [25].

SSN consists of two main parts as shown in Figure 5.1. The first part of the architecture is inspired by [66], whilst the second part that devises the sequence dependencies, is inspired by [29].

- The *epoch encoder* part, or what we refer to as *EPB*, is designed to process 30-second multi-channel EEG epochs, and it aims at learning epoch-wise features. The *EPB* block consists of four modules: (1) spectrogram, (2) signals and frequencies reduction, (3) GRU with attention and (4) positional embedding. In (1) the short-term Fourier transform is computed on each preprocessed epoch, resulting in a time-frequency image $\mathbf{S} \in \mathbb{R}^{C,T,N}$, where C is the number of channels, T is the number of time-steps and N the number of frequency bins. In (2) independent linear projections are applied on the frequencies and channels/signals axis to project $\mathbb{R}^{C,T,N}$ into $\mathbb{R}^{c,T,n}$, where $c \leq C$ and $n \leq N$ are the linearly reduced channels and frequencies respectively. In (3) the reshaped $\mathbb{R}^{T,c,n}$ is the input of the GRU block and the attention layer (implemented as in [100]), and the output is the representation of the sleep epoch in $\mathbb{R}^{2m_1}$, where m_1 are the hidden units of the GRU layer. In (4) they exploit the positional embedding approach recently proposed in [101] to include the whole night PSG context of each epoch in the following sequence encoder block. First, they build a vector $\mathbf{v} = [i_t{}^{\text{epoch}}, i_{t,30}{}^{\text{cycle}}, ..., i_{t,150}{}^{\text{cycle}}] \in \mathbb{R}^6$ for each epoch, where $i_t{}^{\text{epoch}} = \frac{t}{1200}$ is the epoch index and $i_{t,l}{}^{\text{cycle}} = cos(\frac{t\pi}{l})$ with l in $[30, 60, 90, 120, 150]$ are the cyclic indexes. The vector $\mathbf{v}$ is then projected using the Linear+Relu layer to output the positional embedding i_t of each epoch. Finally, i_t is concatenated with the output of the attention layer to obtain the epoch representation $a_t \in \mathbb{R}^{2m_1+6}$.

- The *sequence encoder* part, or what we refer to as *SPB*, is designed to process sequences of epochs, and it aims to encode the temporal information (*e.g.*, stage transition rules). The *SPB* block consists of two layers of bidirectional GRU

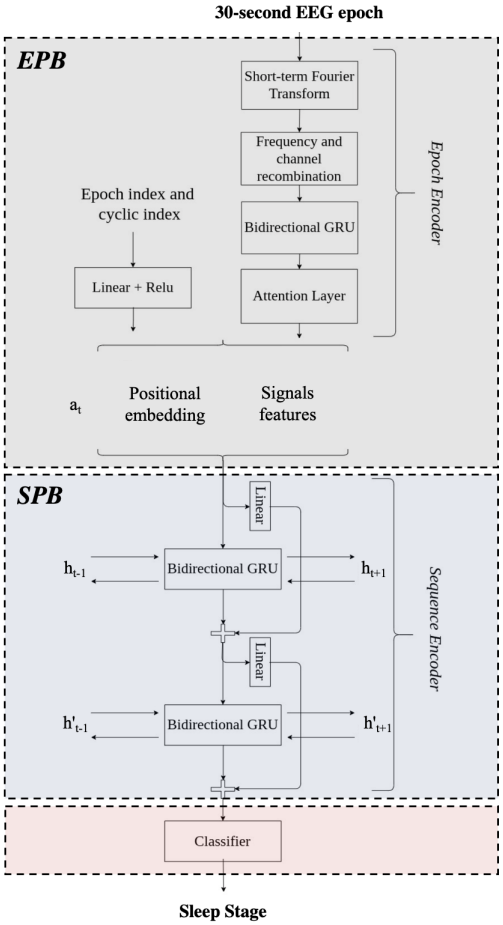


FIGURE 5.1: **SSN architecture.** An overview of the SSN architecture from [25]. h_{t-1}, h'_{t-1} represent the hidden states of the GRU layers from the previous epoch of the sequence and h_{t+1}, h'_{t+1} the hidden states of the GRU layers from the next epoch of the sequence. a_t is the embedding of the current epoch.

with skip-connections (SkipGRU) and the softmax classification layer. The sequence of epochs $a_1, ..., a_t$ is fed to the *SPB* block to output for each epoch the sleep stage probabilities $\hat{p}_k \in \mathbb{R}^5$.

The softmax function, together with the cross-entropy loss function H, is used to train the model to output the probabilities $\hat{p}_k$ for the five mutually exclusive classes K that correspond to the five sleep stages. The architecture is trained end-to-end via backpropagation, using the sequence-to-sequence learning approach. The model is trained using mini-batch Adam gradient-based optimizer [91] with a learning rate lr. The training procedure runs up to a maximum number of iterations (*i.e.*, 100 iterations), as long as the break early stopping condition is satisfied (*i.e.*, the validation F1-score stopped improving for more than 15 epochs; the model with the best validation F1-score is used at test time). All the training parameters (*e.g.*, adam-optimizer parameters beta1 and beta2, mini-batch size, learning rate etc.) are all set as stated in [25].

The architecture has several hyperparameters (*e.g.*, number of layers, number/sizes of filters, regularization parameters, training parameters etc.) which

could be optimized to tune its performance on any dataset. We decided to not systematically tune all these parameters - out of our scope - but to fix them for all the experiments, as done in the original networks.

5.2 Scorer Consensus

Inspired by [25], [67], we evaluate the performance of the sleep scoring architectures, as well as the performance of each physician, using the consensus among the five/six different scorers. The majority vote from the scorers has been computed - *i.e.*, we assign to each 30-second epoch the most voted sleep stage among the physicians. In case of ties, we consider the label from the most reliable scorer. The most reliable scorer is the one that is frequently in agreement with all the others. We use the $Soft - Agreement$ metric proposed in [25] to rank the reliability of each physician, and to finally define the most reliable scorer. We denote with J the total number of scorers and with j the single-scorer. The one-hot encoded sleep stages given by the scorer j are: $\hat{y}_j \in [0,1]^{K \times T}$, where K is the number of classes, *i.e.*, $K = 5$ sleep stages, and T is the total number of epochs. The probabilistic consensus $\hat{z}_j$ among the $J-1$ scorers (j excluded) is computed using the following:

$$\hat{z}_{\mathrm{j}} = \frac{\sum_{i=1}^{J} \hat{y}_{\mathrm{i}}[t]}{max \sum_{i=1}^{J} \hat{y}_{\mathrm{i}}[t]} \qquad \forall t; \quad i \neq j \tag{5.1}$$

where t is the t-th epoch of T epochs and $\hat{z}_j \in [0,1]^{K \times T}$, *i.e.*, 1 is assigned to a stage if it matches the majority or if it is involved in a tie. The $Soft - Agreement$ is then computed across all the T epochs as:

$$Soft - Agreement_{\mathrm{j}} = \frac{1}{T} \sum_{t=0}^{T} \hat{z}_{\mathrm{j}}[y_{\mathrm{j}}] \tag{5.2}$$

where $\hat{z}_{\mathrm{j}}[y_{\mathrm{j}}]$ denotes the probabilistic consensus of the sleep stage chosen by the scorer j for the t-th epoch. $Soft - Agreement_{\mathrm{j}} \in [0,1]$, where the zero value is assigned if the scorer j systematically scores all the annotations incorrectly compared to the others, whilst 1 is assigned if the scorer j is always involved in tie cases or in the majority vote. The $Soft - Agreement$ is computed for all the scorers, and the values are sorted from the highest - high reliability - to the lowest - low reliability. The $Soft - Agreement$ is computed for each patient, *i.e.*, the scorers are ranked for each patient, and in case of a tie the top-1 physician will be the one used for that patient.

5.3 Label smoothing with Soft-Consensus

The predicted sleep stage for each 30-second epoch $x(t)$ comes with a probability value $\hat{p}_i$. As explained in the previous chapters, the probability value associated with the predicted sleep stage should mirror its ground truth correctness likelihood. When this happens, we can state that the model is well calibrated, or that the model provides a calibrated confidence measure along with its prediction [102]. From the previous chapter we also learned that label smoothing [95] has been shown to be a suitable technique to improve the calibration of the model.

By default, the cross-entropy loss function is computed between the prediction $\mathbf{p}_{\mathrm{i}}$ and the target $\mathbf{y}_{\mathrm{i}}$ (*i.e.*, the one-hot encoded sleep stages, 1 for the correct class

and 0 for all the other classes). When the model is trained with the label smoothing technique, the hard target is smoothed with the standard uniform distribution $1/K$ (eq. 5.3). Thus, the cross-entropy loss function (eq. 5.4) is minimized by using the weighted mixture of the target ${y_{i,k}}^{LS_U}$.

$${y_{i,k}}^{LS_U} = y_{i,k} \cdot (1-\alpha) + \alpha/K \tag{5.3}$$

$$H(\mathbf{y}_i, \mathbf{p}_i) = \sum_{k=1}^{K} -{y_{i,k}}^{LS_U} \cdot \log(\hat{p}_{i,k}) \tag{5.4}$$

where α is the smoothing parameter, K is the number of sleep stages, ${y_{i,k}}^{LS_U}$ the weighted mixture of the target and $\hat{p}_{i,k}$ the output of the softmax layer with the predicted probability values.

In our experiments, we exploit the label smoothing technique to insert the knowledge from the multiple-scorers in the learning process. We propose to use the $Soft-Consensus$ (eq. 5.5) as our new distribution to smooth the hard target $y_{i,k}$.

$$Soft-Consensus_i = \frac{\#(Y_i = y_{i,k})}{M} \tag{5.5}$$

where Y_i is the set of observations - *i.e.*, annotations given by the different physicians - for the i-th epoch, k is the class index, M is the number of observations and # is the cardinality of the set $(Y_i = y_{i,k})$. In simple words, the probability value for each sleep stage k is computed as the sum of its occurrences divided by the total number of observations.

$Soft-Consensus_i \in [0,1]^{1\times K}$ is the one-dimensional vector that we use to smooth the hard target (eq. 5.6), and then minimize the cross-entropy loss function (eq. 5.7).

$${y_{i,k}}^{LS_{SC}} = y_{i,k} \cdot (1-\alpha) + \alpha \cdot Soft-Consensus_{i,k} \tag{5.6}$$

$$H(\mathbf{y}_i, \mathbf{p}_i) = \sum_{k=1}^{K} -{y_{i,k}}^{LS_{SC}} \cdot \log(\hat{p}_{i,k}) \tag{5.7}$$

As an example, consider the following set of observations $Y_i = [W, W, W, N1, N2]$ given by five different physicians for the i-th epoch. By applying (eq. 5.5) and (eq. 5.6) we obtain the following ${y_{i,k}}^{LS_{SC}}$ smoothed hard-target:

$$Soft-Consensus_{i,k} = [\hat{p}_W = 3/5, \hat{p}_{N1} = 1/5, \hat{p}_{N2} = 1/5, \hat{p}_{N3} = 0/5, \hat{p}_R = 0/5]$$

$$Soft-Consensus_{i,k} = [0.6, 0.2, 0.2, 0, 0]$$

$${y_{i,k}}^{LS_{SC}} = y_{i,k} \cdot (1-\alpha) + \alpha \cdot Soft-Consensus_{i,k} = [0.8, 0.1, 0.1, 0, 0]$$

with the one-hot encoded target $y_{i,k} = [1, 0, 0, 0, 0]$ and $\alpha = 0.5$.

We perform a simple grid-search to set the smoothing hyperparameter α. When the model is trained with the labels smoothed by the uniform distribution $\alpha \in (0, 0.5]$ with step 0.1. Extreme values are not considered as for $\alpha = 0$ the model is trained using the standard hot-encoding vector; whilst for values higher than 0.5, *e.g.*, $\alpha = 1$, the model would be trained using mainly/only the uniform distribution $1/K$ for

each sleep stage. When the model is trained with the labels smoothed by the $Soft-Consensus$ distribution $\alpha \in (0, 1]$ with step 0.1. In the latter case we also investigate an α value equal to 1 to evaluate the full impact of the consensus distribution on the learning procedure.

5.4 Databases

We use the IS-RC (Inter-scorer Reliability Cohort) introduced in [99] to assess the inter-scorer reliability among different sleep centers and the two publicly available databases introduced in [25] DOD-H (Dreem Open Dataset - Healthy) and DOD-O (Dreem Open Dataset - Obstructive).

IS-RC. The dataset contains 70 recordings (0 males and 70 females) from patients with sleep-disordered breathing aged from 40 to 57. The recordings were collected at the University of Pennsylvania. Each recording includes the EEG derivations C3-M2, C4-M1, O1-M2, O2-M1, one EMG channel, left/right EOG channels, one ECG channel, nasal airway pressure, oronasal thermistor, body position, oxygen saturation and abdominal excursion. The recordings are sampled at 128 Hz. We only consider the single-channel EEG C4-M1 to train our DSN-L architecture, and we use multi-channel EEG, EOG, EMG and ECG to train the SSN architecture. A band-pass Chebyshev IIR filter is applied between [0.3, 35] Hz. Each recording is scored by six clinicians from five different sleep centers (*i.e.*, University of Pennsylvania, University of Wisconsin at Madison, St. Luke's Hospital (Chesterfield), Stanford University and Harvard University) according to the AASM rules [2]. The dataset contains the following annotations W, $N1$, $N2$, $N2$, R, and NC, where NC is a not classified epoch. Some epochs are not scored by all the six physicians, and even for some of them we don't have any annotation (*i.e*, NC). We decided to remove the epochs classified by all the scorers as NC. Epochs with less than six annotations are equally taken into account to avoid excessive data loss.

DOD-H. The dataset contains 25 recordings (19 males and 6 females) from healthy adult volunteers aged from 18 to 65 years. The recordings were collected at the French Armed Forces Biomedical Research Institute's (IRBA) Fatigue and Vigilance Unit (Bretigny-Sur-Orge, France). Each recording includes the EEG derivations C3-M2, C4-M1, F3-F4, F3-M2, F3-O1, F4-O2, O1-M2, O2-M1, one EMG channel, left/right EOG channels and one ECG channel. The recordings are sampled at 512 Hz. **DOD-O**. The dataset contains 55 recordings (35 males and 20 females) from patients suffering from obstructive sleep apnea (OSA) aged from 39 to 62 years. The recordings were collected at the Stanford Sleep Medicine Center. Each recording includes the EEG derivations C3-M2, C4-M1, F4-M1, F3-F4, F3-M2, F3-O1, F4-O2, FP1-F3, FP1-M2, FP1-O1, FP2-F4, FP2-M1, FP2-O2, one EMG channel, left/right EOG channels and one ECG channel. The recordings are sampled at 250 Hz. We only consider the single-channel EEG C4-M1 to train our DSN-L architecture, and we use all the available channels to train SSN architecture, on both DOD-H and DOD-O. As in [25], a band-pass Butterworth IIR filter is applied between [0.4, 18] Hz to remove residual PSG noise, and the signals are resampled at 100 Hz. The signals are then clipped and divided by 500 to remove extreme values. The recordings from both DOD-H and DOD-O datasets are scored by five physicians from three different sleep centers according to the AASM rules [2]. DOD-H and DOD-O contain the following annotations W, $N1$, $N2$, $N3$, R, and NC, where NC is a not classified

TABLE 5.1: **Sleep stages on IS-RC, DOD-H and DOD-O.**
Number and percentage of 30-second epochs per sleep stage for the IS-RC, DOD-H and DOD-O datasets.

Datasets	W	N1	N2	N3	R	Total
IS-RC	24517 (29.1%)	3773 (4.5%)	40867 (48.5%)	3699 (4.4%)	11475 (13.6%)	84331
DOD-H	3075 (12.5%)	1463 (5.9%)	12000 (48.7%)	3442 (14.0%)	4685 (19.0%)	24665
DOD-O	10520 (19.8%)	2739 (5.1%)	26213 (49.2%)	5617 (10.6%)	8147 (15.3%)	53236

epoch. All the scorers agree about the *NC* epochs (100% of agreement). Therefore, all of them are removed from the data. Unlike the previous IS-RC database, for each epoch five annotations are always available.

In Table 5.1 we report a summary of the total number and percentage of the epochs per sleep stage for the DOD-H, DOD-O and IS-RC datasets.

TABLE 5.2: **Data split on the IS-RC, DOD-H and DOD-O datasets.**

Datasets	Size	Experimental Setup	Held-out Validation Set	Held-out Test Set
IS-RC	70	10-fold CV	13 subjects	7 subject
DOD-H	25	25-fold CV	6 subjects	1 subjects
DOD-O	55	10-fold CV	12 subjects	6 subjects

5.5 Results

5.5.1 Experiment Designs

We evaluate DSN-L and SSN using the k-fold cross-validation scheme. We set k equal to 10 for IS-RC, 25 for DOD-H (leave-one-out evaluation procedure) and 10 for DOD-O datasets. In Table 5.2 we summarize the data split for each dataset.

The following experiments are conducted on both DSN-L and SSN models for each dataset:

- ***base***. The models are trained without label smoothing.
- ***base+LS*$_{\mathbf{U}}$**. The models are trained with label smoothing using the standard $1/K$ uniform distribution - *i.e.*, the hard targets (scorer consensus) are weighted with the uniform distribution.
- ***base+LS*$_{\mathbf{SC}}$**. The models are trained with label smoothing using the proposed $Soft-Consensus$ - *i.e.*, the hard targets (scorer consensus) are weighted with the soft-consensus distribution.

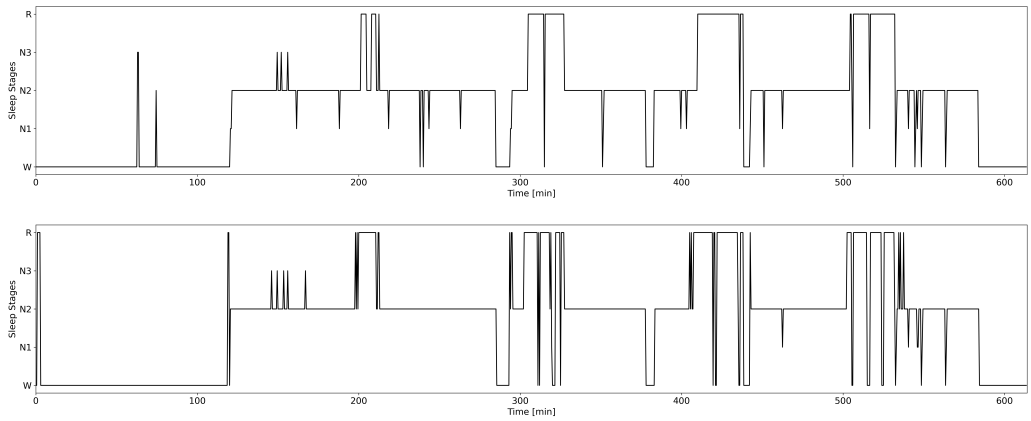


FIGURE 5.2: **Hypnogram.** Discrete sleep stage values for each 30-second epoch of a patient from IS-RC. (top) Majority vote from the scorers labels. (bottom) Predicted labels from the DSN-L *base+LS*$_{SC}$ based model.

These models, differently trained, have been evaluated with and without using the *MC* dropout ensemble technique. In Tables 5.4 and 5.5 subsection 5.5.3 we present the results obtained for each experiment on both DSN-L and SSN evaluated on IS-RC, DOD-H and DOD-O datasets.

5.5.2 Metrics

The per-class F1-score, the overall accuracy (*Acc.*), the macro-averaging F1-score (*MF1*), the weighted-averaging F1-score (*F1*) - *i.e.*, the metric is weighted by the number of true instances for each label, so as to consider the high imbalance between the sleep stages - and the Cohen's kappa (k) have been computed from the predicted sleep stages from all the folds to evaluate the performance of our model [88], [89].

Model Calibration. We evaluated the calibration of our model using the standard ECE value as in subsection 4.5.2. Perfectly calibrated models have $acc(B_m) = conf(B_m)$ for all $m \in \{1, .., M\}$, resulting in $ECE = 0$ (eq. 4.14).

Hypnodensity-graph. The hypnodensity-graph is an efficient visualization tool introduced in [67] to plot the probability distribution over each sleep stage for each 30-second epoch over the whole night. Unlike the standard hypnogram sleep cycle visualization tool, it shows the probability of occurrence of each sleep stage for each 30-second epoch; so it is not limited to the discrete sleep stage value (see Figure 5.2 and Figure 5.3). In our analysis we have used the hypnodensity-graph to display both the model output - *i.e.*, the softmax output probability vectors $\hat{p}_{i,k}$ - and the multi-scorer $Soft - Consensus_{i,k}$ probability distributions.

The Averaged Cosine Similarity (ACS) is used to quantify the similarity between the hypnodensity-graph generated by the model and the hypnodensity-graph generated by the $Soft - Consensus$. The ACS has been computed as follows:

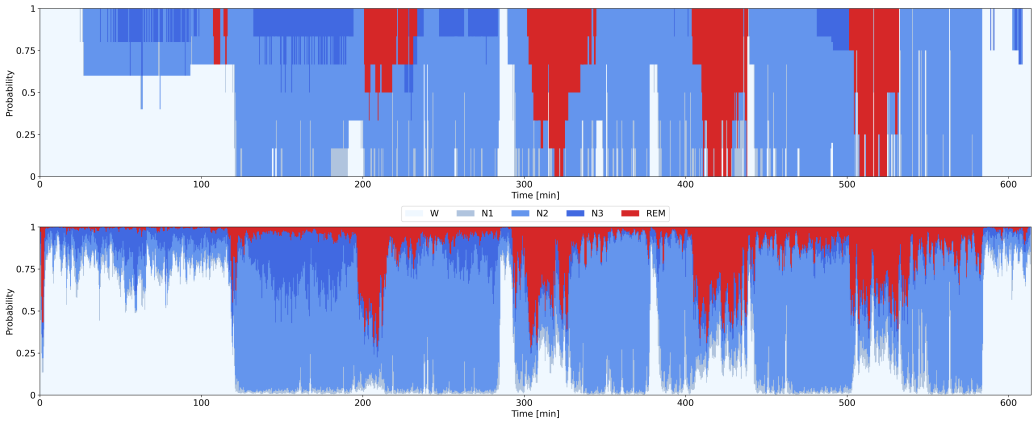


FIGURE 5.3: **Hypnodensity-graph.** Cumulative probabilities of each sleep stage for each 30-second epoch of a patient from IS-RC. (top) Soft-consensus from the scorers labels. (bottom) Predicted probabilities from the DSN-L *base+LS*$_{SC}$ based model.

$$ACS = \frac{1}{N}\sum_{i=1}^{N}\frac{Soft-Consensus_{i,k}\cdot\hat{p}_{i,k}}{\parallel Soft-Consensus_{i,k}\parallel\cdot\parallel\hat{p}_{i,k}\parallel} \tag{5.8}$$

where N is the number of epochs in the whole night, $\parallel . \parallel$ is the norm computed for the predicted probability vector $\hat{p}_{i,k}$ and the $Soft-Consensus_{i,k}$ ground-truth vector for the i-th epoch. Thus, the cosine-similarity is averaged across all the epochs N to obtain our averaged ACS unique score of similarity. The cosine-similarity values may range between 0, *i.e.*, high dissimilarity, and 1, *i.e.*, high similarity between the vectors.

5.5.3 Analysis of Experiments

In Table 5.3 we first report for all the multi-scored databases IS-RC, DOD-H and DOD-O, the overall scorers performance and their $Soft-Agreement$ (SA), *i.e.*, the agreement of each scorer with the consensus among the physicians. On IS-RC we have a lower inter-scorer agreement (*i.e.*, SA equal to 0.69 and F1-score 69.7%) compared to both DOD-H and DOD-O (*i.e.*, SA equal to 0.89 and 0.88, with an F1-score 88.1% and 86.4% respectively). Consequently, we expect a higher efficiency of our label smoothing with soft-consensus approach (*base+LS*$_{SC}$) on the experiments conducted on the IS-RC database. The lower the inter-scorer agreement, the lower the performance of a model trained with the one-hot encoded labels (*i.e.*, the majority vote weighted by the degree of consensus from each physician).

In Tables 5.4 and 5.5 we report the overall performance, the calibration measure and the hypnodensity similarity measures of the three different DSN-L and SSN models on the three databases IS-RC, DOD-H and DOD-O. The performance of the DSN-L *base* models are higher compared to the performance averaged among the scorers on the IS-RC database, but not on the DOD-H and DOD-O databases. In

TABLE 5.3: **Scorers performance.** Scorers performance on IS-RC, DOD-H and DOD-O datasets with $Soft - Agreement$ (SA), overall accuracy (%Acc.), macro F1-score (%MF1), Cohen's Kappa (k), weighted-averaging F1-score (%F1) and % per-class F1-score. The scorer with the best performance (*i.e.*, high agreement with the consensus among the six different physicians) is indicated in bold.

			Overall Metrics				Per-Class F1-Score				
Datasets	Scorers	SA	Acc.	MF1	k	F1	W	N1	N2	N3	R
IS-RC	Scorer-1	0.79	82.9	69.4	0.72	83.7	82.5	47.2	87.3	48.1	89.9
	Scorer-2	0.81	89.3	72.6	0.82	89.1	90.7	57.7	92.5	32.4	89.9
	Scorer-3	0.53	40.8	26.5	0.11	40.9	29.5	14.7	54.7	18.1	15.4
	Scorer-4	0.52	38.9	26.0	0.12	40.6	28.3	14.6	54.3	15.5	17.3
	Scorer-5	0.70	73.7	61.5	0.63	75.8	88.6	36.7	70.3	25.8	86.3
	Scorer-6	0.79	87.2	77.1	0.81	88.2	92.5	54.3	89.4	59.8	89.6
	Average	0.69	68.8	55.5	0.53	69.7	68.7	37.5	74.8	33.3	64.7
DOD-H	Scorer-1	0.88	82.3	77.1	0.75	83.1	86.3	44.0	85.7	83.6	85.9
	Scorer-2	0.91	83.7	78.5	0.77	84.3	87.4	47.9	87.0	82.7	87.6
	Scorer-3	0.92	84.5	79.5	0.78	84.9	87.9	50.6	87.4	82.0	89.4
	Scorer-4	0.84	83.1	77.6	0.76	83.5	83.3	47.6	86.2	81.5	89.6
	Scorer-5	0.92	83.7	78.2	0.77	83.9	83.8	48.5	86.8	81.8	89.8
	Average	0.89	83.5	78.2	0.76	83.9	85.7	47.7	86.6	82.3	88.5
DOD-O	Scorer-1	0.87	80.7	72.6	0.72	80.1	87.6	38.3	83.2	67.1	86.9
	Scorer-2	0.87	80.7	73.6	0.72	80.7	87.7	41.6	83.0	67.3	86.9
	Scorer-3	0.88	80.9	73.5	0.72	80.8	88.0	41.7	83.3	66.0	88.7
	Scorer-4	0.88	81.2	74.0	0.73	81.3	88.2	42.7	83.9	66.3	88.7
	Scorer-5	0.91	81.8	74.7	0.74	82.0	88.5	43.8	84.5	67.5	89.0
	Average	0.88	81.1	73.7	0.73	81.0	88.0	41.6	83.6	66.8	88.3

contrast, the performance of the SSN *base* models are always higher than the performance averaged among the scorers on all the databases. We highlight that the results we report for SSN on DOD-H and DOD-O are slightly different compared to the one reported in [25]. We decided to not compute a weight (from 0 to 1) for each epoch, based on how many scorers voted for the consensus. We do not balance the importance of each epoch when we compute the above mentioned metrics. We think it is unfair to constrain any metrics based on the amount of voting physicians. Overall, the results show a significant improvement in performance on all the databases (i.e overall accuracy, MF1-score, Cohen's kappa (k) and F1-score) from the baseline *base* and the label smoothing with the uniform distribution (*base*+LS_{U}) models, to the ones trained with label smoothing along with the proposed $Soft - Consensus$ distribution (*i.e.*, *base*+LS_{SC}). The ACS is the metric that best quantifies the ability of the model in adapting to the consensus of the group of scorers. A higher ACS value means a higher similarity between the hypnodensity-graph generated by the model and the hypnodensity-graph generated by the $Soft - Consensus$ (*i.e.*, the model better adapts to the consensus of the group of physicians). As all the other metrics the ACS value is computed per subject, but here we report the mean and also the standard deviation across subjects ($\mu \pm \sigma$). We found a significant improvement in the ACS value from the *base* and the *base*+LS_{U} models to the

TABLE 5.4: **DSN-L models performance +LS_{SC}.** Overall metrics, per-class F1-score, calibration and *ACS* hypnodensity-graph similarity measures of the DSN-L models obtained from 10-fold cross-validation on IS-RC dataset, from 25-fold cross-validation on DOD-H dataset, and from 10-fold cross-validation on DOD-O dataset. Best shown in bold.

			Overall Metrics					Per-Class F1-Score				Calibration		Hypn.
Datasets	Models	α	Acc.	MF1	k	F1	W	N1	N2	N3	R	ECE	*conf*	*ACS*
IS-RC	*base*	-	69.6	50.6	0.56	70.0	81.6	11.8	71.9	27.2	60.7	**0.096**	79.0	0.772 ± 0.075
	base+LS_U	0.4	74.8	**57.0**	0.63	75.8	83.3	**24.3**	79.0	30.6	**67.7**	0.296	45.2	0.806 ± 0.042
	base+LS_{SC}	0.6	**75.8**	56.5	**0.64**	**75.9**	**83.5**	19.5	**79.7**	**33.3**	66.4	0.190	56.7	$\mathbf{0.836 \pm 0.041}$
DOD-H	*base*	-	76.9	70.0	0.68	77.2	79.7	39.5	78.8	76.5	75.2	0.163	92.7	0.817 ± 0.097
	base+LS_U	0.2	75.3	68.7	0.66	75.2	78.8	40.0	75.9	72.0	76.8	0.059	68.9	0.829 ± 0.068
	base+LS_{SC}	0.8	**80.2**	**72.4**	**0.72**	**80.4**	**80.4**	**42.3**	**83.4**	**77.6**	**78.4**	**0.016**	81.4	$\mathbf{0.873 \pm 0.053}$
DOD-O	*base*	-	77.3	67.8	0.66	78.0	80.7	41.2	81.0	68.1	68.3	0.131	90.2	0.840 ± 0.073
	base+LS_U	0.1	77.5	68.0	0.67	78.2	**80.8**	41.9	80.4	68.4	**68.7**	0.009	78.4	0.859 ± 0.072
	base+LS_{SC}	1	**79.4**	**69.6**	**0.69**	**79.9**	80.4	**43.8**	**83.5**	**72.5**	68.1	**0.009**	78.3	$\mathbf{0.878 \pm 0.061}$

TABLE 5.5: **SSN models performance +LS_{SC}.** Overall performance, calibration and *ACS* hypnodensity-graph similarity measures of the SSN models obtained from 10-fold cross-validation on IS-RC dataset, from 25-fold cross-validation on DOD-H dataset, and from 10-fold cross-validation on DOD-O dataset. Best shown in bold.

			Overall Metrics					Per-Class F1-Score				Calibration		Hypn.
Datasets	Models	α	Acc.	MF1	k	F1	W	N1	N2	N3	R	ECE	*conf*	*ACS*
IS-RC	*base*	-	81.8	**60.8**	0.72	80.8	86.3	**29.9**	85.3	**24.3**	78.1	0.174	99.4	0.806 ± 0.052
	base+LS_U	0.3	82.5	59.8	0.72	81.1	86.5	28.8	86.5	18.7	78.7	0.169	99.3	0.811 ± 0.058
	base+LS_{SC}	0.7	**83.1**	60.2	**0.73**	**81.6**	**86.7**	27.6	**86.8**	20.1	**79.8**	**0.162**	99.2	$\mathbf{0.817 \pm 0.047}$
DOD-H	*base*	-	87.1	80.2	0.81	87.1	83.6	55.5	90.0	**83.3**	89.0	0.126	99.7	0.890 ± 0.047
	base+LS_U	0.4	87.6	81.0	0.81	87.5	85.5	57.3	90.2	82.1	90.3	0.120	99.5	0.899 ± 0.034
	base+LS_{SC}	0.5	**88.8**	**82.3**	**0.83**	**88.7**	**86.4**	**58.8**	**90.9**	83.2	**92.1**	**0.108**	99.6	$\mathbf{0.907 \pm 0.039}$
DOD-O	*base*	-	85.3	75.9	0.77	85.2	88.2	50.4	87.1	65.9	88.0	0.145	99.7	0.889 ± 0.056
	base+LS_U	0.1	85.6	75.8	0.78	85.2	88.2	**51.2**	87.3	64.3	88.4	0.141	99.6	0.893 ± 0.052
	base+LS_{SC}	1	**86.8**	**77.7**	**0.79**	**86.7**	**89.0**	51.0	**88.3**	**69.3**	**91.1**	**0.125**	99.2	$\mathbf{0.906 \pm 0.043}$

base+LS_{SC} models on all the databases and on both DSN-L ($p - values < 0.01$) and SSN ($p - values < 0.05$). Hence, our approach enables both the DSN-L and the SSN architectures to significantly adapt to the consensus of the group of physicians on all the multi-scored datasets.

We could easily infer that the SSN architecture is better (*i.e.*, higher performance) compared to our DSN-L architecture. The purpose of our study is not to highlight whether one architecture is better than the other, but we can not fail to notice the high values of confidence (the $conf$ value is the average of the softmax output max-probabilities) obtained on the SSN based models. High values of confidence still persist despite smoothing the labels (with both uniform and soft-consensus distributions) during the training procedure. The SSN architecture is not highly responsive to the changes in probability values we implemented on the one-hot encoded labels. It always rely/overfit on the max probability value given for each epoch, i.e. the consensus among the five/six different scorers. Indeed, on the IS-RC, which is the database with the lower inter-scorer agreement, the SSN *base*+LS_{SC} model reaches a higher value of F1-score, i.e. 81.6%, compared to our DSN-L *base*+LS_{SC} model,

TABLE 5.6: **DSN-L and SSN models performance +LS_{SC} w/ and w/o *MC*.** Overall metrics and *ACS* hypnodensity-graph similarity measures on the DSN-L and SSN *base*+LS_{SC} models, obtained from 10-fold cross-validation on IS-RC dataset, from 25-fold cross-validation on DOD-H dataset, and from 10-fold cross-validation on DOD-O dataset with and without *MC*. Best shown in bold.

			Overall Metrics				Hypn.
Datasets	Model	*MC*	Acc.	MF1	k	F1	*ACS*
IS-RC	DSN-L	*w/o*	75.8	56.5	0.69	75.9	0.836 ± 0.041
		w/	**78.6**	**57.6**	**0.67**	**78.0**	**0.850 ± 0.036**
	SSN	*w/o*	**83.1**	**60.2**	0.73	**81.6**	0.817 ± 0.047
		w/	83.0	59.2	**0.73**	81.1	**0.818 ± 0.048**
DOD-H	DSN-L	*w/o*	80.2	72.4	0.72	80.4	0.873 ± 0.053
		w/	**84.4**	**75.9**	**0.76**	**84.2**	**0.906 ± 0.026**
	SSN	*w/o*	88.8	82.3	0.83	88.7	0.907 ± 0.039
		w/	**89.1**	**82.6**	**0.84**	**89.0**	**0.910 ± 0.039**
DOD-O	DSN-L	*w/o*	79.4	69.6	0.69	79.9	0.878 ± 0.061
		w/	**80.7**	**70.8**	**0.71**	**80.9**	**0.889 ± 0.059**
	SSN	*w/o*	86.8	77.7	0.79	86.7	0.906 ± 0.043
		w/	**87.1**	**78.0**	**0.80**	**86.9**	**0.909 ± 0.041**

i.e. 75.9% , but a lower value of *ACS* (0.817 on SSN and 0.836 on DSN-L, with a $p - value < 0.01$). The SSN model overfit to the majority vote or the *max* probability value given for each epoch, whilst the DSN-L better adapts to the consensus of the group of scorers (*i.e.*, better encode the variability among the physicians).
The last statement is also strengthened by Supplementary Figure B.8 and Supplementary Figure B.9 in Appendix B. For DSN-L and SSN we report the *ACS* values across all the experimented values, on both the *base*+LS_U and the *base*+LS_{SC} models tested on the three databases. As expected, the DSN-L model shows a high sensitivity in *ACS* values to changes in α-hyperparameter across all databases. This sensitivity is not as strong with the SSN model.

Moreover, we want to highlight that the standard uniform distribution is not as efficient as the proposed soft-consensus distribution in encoding the scorer's variability. By using the uniform distribution we are not able to learn as well the complexity of the degree of agreement between the different physicians. Indeed, in Supplementary Figure B.8, on the DSN-L model, we clearly show how the *ACS* value proportionally increase with the α-hyperparameter only by using the proposed soft-consensus distribution.

In our study we exploit, as in Chapter 4, the *Monte Carlo* (*MC*) *dropout* ensembling technique to further enhance the performance of the models. We apply *MC* dropout $M = 30$ times at inference time. The final prediction $\hat{y}_i$ will be given by $max(\boldsymbol{\mu}_i)$, which we will refer to as $\mu_{\max}$, along with the assigned variance value ${\sigma^2}_{\mu\max}$. In Table 5.6 we show that, overall, on all the experiments, we obtained a slight improvement (up to 4%) on our best *base*+LS_{SC} models, on IS-RC, DOD-H and DOD-O datasets.

5.5.4 Uncertainty estimate

In Chapter 4 we also introduced a query procedure to estimate the uncertain predictions given by the model. The query procedure simply relies on the setting of a fixed threshold value q%=5%, that corresponds to a percentage of sleep epochs to select/reject and to send potentially to the physician for a secondary review. The epochs with the lowest probability values are the q% selected (on average up to 50 epochs for each PSG recording). In this study we simply use baseline/standard query selection procedure, *i.e.*, *w/o MC* the predicted probability values $max(\hat{\boldsymbol{p}}_{\mathrm{i}})$ are directly used to select/reject the uncertain sleep epochs.

In Supplementary Table B.4 we report the overall performance achieved on the DSN-L models on IS-RC, DOD-H and DOD-O datasets as a result of the baseline/standard query procedure. Specifically, the metrics refer to the epochs kept after the $max(\hat{\boldsymbol{p}}_{\mathrm{i}})$ base selection procedures (q% threshold value fixed to 5%). We quantify the percentage of misclassified epochs (%miscl.) among the rejected after the query procedure. The percentage of misclassified epochs is on average in the range 50% to 60%. Consequently, on all the models we have an increase in performance up to 2%-3% in F1-score. These results highlight the efficiency of the query procedure to select a good enough number of misclassified epochs among the selected one. Unlike what we expected, it is not always the case that a better calibrated architecture leads to a better estimate of the model uncertainty. A lower ECE value (see Table 5.4) does not always enable the detection of a significantly higher number of percentages of misclassified epochs (%miscl.).

5.6 Discussion

In this Chapter we demonstrate the efficiency of label smoothing along with the soft-consensus distribution in encoding the scorers's variability into the training procedure of a sleep scoring algorithm, specifically on both DSN-L and SSN architectures. The results show an improvement in overall performance from the *base* and the *base*+LS_{U} models to the ones trained with *base*+LS_{SC}. We introduce the averaged cosine similarity metric to better quantify the similarity between the probability distribution predicted by the models and the ones generated by the scorer consensus. We obtained a significant improvement in the *ACS* values with our *base*+LS_{SC} models on both DSN-L and SSN architectures. Based on the reported high confidence values, we found that SSN tends to overfit on each dataset. Specifically, it tends to overfit on the majority vote weighted by the degree of consensus from each physician, consequentially it does not encode as well their variability.

The proposed procedure is quite simple and it enables us to transfer the variability, the uncertainty, and the noise we have by nature on the sleep labels into the models. Our approach results quite effective in encoding the complexity of the scorers' consensus within the classification algorithm, whose importance is often underestimated. Moreover, by leveraging both the LS_{SC} technique and the uncertainty estimate procedure described above, we are able to increase the percentage of detected misclassified epochs among those discarded/removed (up to 60%). The present approach enables us to better adapt to the consensus of the group of scorers, and, as a consequence, to better quantify the uncertainty and the disagreement we have between the different scorers.

Chapter 6

U-Sleep: resilient to AASM guidelines

In literature we find many examples about how clinical guidelines have been exploited into trying to support ML and DL based algorithms. The oldest R&K [9] or the updated AASM [2] scoring manuals have been designed to cover all the aspects of the polysomnography: from the technical/digital specifications (*e.g.*, assessment protocols, data filtering, recommended EEG derivations) to the scoring rules (*e.g.*, sleep scoring rules for adults, children and infants, movement rules, respiratory rules) and the final interpretation of the results. To date, all the sleep scoring algorithms, both ML or DL based, are trained on sleep recordings annotated by sleep physicians according to these manuals. For example, in [67] they pre-filter the sleep recordings as indicated in the AASM guidelines before feeding them to their scoring system; almost all of the algorithms mentioned above were trained using recommended channel derivations and fixed length (*i.e.*, 30-second) sleep epochs. However, it still remains unknown whether a DL based sleep scoring algorithm actually needs to be trained by following these guidelines. A decade ago, it was already highlighted that sleep is not a global phenomenon affecting the whole brain at the same time, and that sleep patterns, such as slow waves and spindle oscillations, often occur out-of-phase in different brain regions [103], [104]. Hence, the usage of multi-channel derivations, but not necessarily the ones indicated in the AASM guidelines, may be enough to reach high performance with our DL based scoring algorithms. Furthermore, in the AASM manual and in previous studies [105], [106], age has been addressed as one of the demographic factor that mainly change sleep characteristics (*e.g.*, sleep latency, sleep cycle structure, EEG amplitude etc.). To the best of our knowledge, it has never been attempted before to incorporate this information within a sleep scoring system; it could reasonably improve its performance.

To date, all the effort has focused on optimizing a sleep scoring algorithm in order to be ready to score any kind of subject. Data mismatch and data heterogeneity is one of the biggest challenges to address. The performance of a sleep scoring algorithm on a PSG from an unseen data distribution (*e.g.*, different data domain/center) usually drastically decreases. A common objective among researchers is to increase the model generalizability, *i.e.*, the ability of the model to make accurate predictions over different or never seen data domains. In recent studies, [71], [107] propose to adapt a sleep scoring architecture on a new data domain via transfer learning techniques. They demonstrate the efficiency of their approaches in addressing the variability between the source and target data domains. [57], [108], [109] propose to train their sleep scoring architectures on tens of thousands of PSGs from different large-scale-heterogeneous cohorts. They demonstrate that using data from many different sleep centers improves the performance of their model, even on never

seen data domains. In particular, [108] shows that models trained on a single data domain fail to generalize on a new data domain or data center.

In Chapter 6 we did several experiments to evaluate the resilience of an existing DL based algorithm against the AASM guidelines. In particular we focused on the following questions: *(i)* can a sleep scoring algorithm successfully encode sleep patterns, from clinically non-recommended derivations? *(ii)* can a single sleep center large dataset contain enough heterogeneity (*i.e.*, different demographic groups, different sleep disorders) to allow the algorithm to generalize on multiple data centers? *(iii)* whenever we train an algorithm on a dataset with subjects with a large age range, should we exploit the information about their age, conditioning the training of the model on it?

To address this set of questions, we run all our experiments on U-Sleep, a state-of-the-art sleep scoring architecture recently proposed in [57]. U-Sleep has been chosen mainly for the following reasons: it has been evaluated on recordings from 15660 participants of 16 different clinical studies (four of them never seen by the architecture); it processes inputs of arbitrary length, from any arbitrary EEG and EOG electrode positions, from any hardware and software filtering; it predicts the sleep stages for the whole PSG in a single forward pass; it outputs sleep stage labels at any temporal frequency, up to the signal sampling rate, *i.e.*, it can label sleep stages at shorter intervals than the standard 30-seconds, up to one sleep stage per each sampled time point.

In the original implementation of U-Sleep we found an extremely interesting *bug*: the data sampling procedure was not extracting the channel derivations recommended in the AASM guidelines, as stated by the authors in [57]. Instead, atypical or non-conventional channel derivations were randomly extracted. This insight triggered the above mentioned question *(i)*.

In the previous chapters we evaluated our uncertainty estimate procedure on existing sleep scoring architectures on small-sized databases. In Chapter 6 we also want to investigate the efficiency of the uncertainty estimate procedure, together with the model ensembling and the label smoothing techniques, on U-Sleep architecture, evaluated on multiple and large-scale databases.

Contributions. Our contributions can be summarized as follows: (1) we find that a DL based scoring algorithm is still able to solve the scoring task, with high performance, even when trained with clinically non-conventional channel derivations; (2) we show that a sleep scoring model, even if trained on a single large and heterogeneous data domain, fails to generalize on new recordings from different data centers; (3) we demonstrate that the conditional training based on the chronological age of the subjects is unnecessary; (4) we prove the efficiency of the uncertainty estimate procedure in detecting a relevant number of challenging epochs, even when evaluated on multiple large-scale databases.

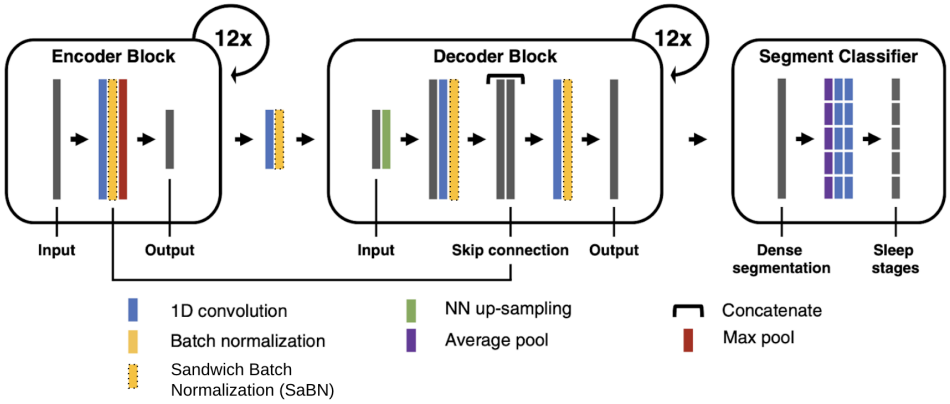


FIGURE 6.1: **U-Sleep overall architecture.** [57]. We also report the additional Sandwich Batch Normalization layers exploited in the conditional learning procedure (see subsection 6.3). Please refer to [57] for details on the U-Sleep model architecture and training parameters.

6.1 U-Sleep

U-Sleep [57], optimized version of its predecessor U-Time [56], is inspired by the popular U-Net architecture for image-segmentation [110]–[112]. Below we briefly describe U-Sleep architecture, for further details we refer the reader to [57].

6.1.1 Architecture

U-Sleep is a fully convolutional deep neural network. It takes as input a sequence of length L of 30-second epochs and outputs the predicted sleep stage for each epoch. The peculiarity of this architecture is that it defines the general function $f(\mathbf{X};\theta) : \mathbb{R}^{L\cdot i\times C} \rightarrow \mathbb{R}^{L\times K}$, where $L > 0$ is any positive integer, θ are the learning parameters, L is a number of fixed-length windows with i sampled points each, C the number of PSG channels and K the number of sleep stages. Hence, U-Sleep takes in input any temporal section of a PSG (even the whole PSG) and output a sequence of labels for each fixed-length $i > 0$ window. Ideally $L \cdot i > 4096$, because U-Sleep contains 12 pooling operations, downsampling the signal by a factor of 2. The architecture requires at least $C = 2$, one EEG and one EOG channel, sampled/resampled at 128Hz, with $K = 5$, *i.e.*, *awake*, $N1$, $N2$, $N3$, R.

U-Sleep architecture consists of three learning modules as shown in Figure 6.1.

- The *encoder* module is designed to extract feature maps from the input signals, each resulting in a lower temporal resolution compared to its input. The module includes 12 encoder blocks. Each block consists of a 1D convolutional layer, one layer of activation function - *i.e.*, exponential linear unit (ELU), a batch normalization layer and one max-pooling layer.

- The *decoder* module is designed to up-scale the feature maps to match the temporal resolution of the signals in input. We can interpret the output of the decoder as a high-frequency representation of the sleep stages at the same f_s of the input signal (*e.g.*, with $f_s = 128Hz$, output one sleep stage each $1/128Hz$). The module includes 12 decoder blocks. Each block consists of a nearest neighbour up-sampling layer (*e.g.*, with a $kernel\_size = 2$, the length of the feature

map in input is doubled), a 1D convolutional layer, one layer of ELU activation function and a batch normalization layer. Then, a skip connection layer combines the up-scaled input with the output of the batch normalization layer of the corresponding encoder block. Finally, a 1D convolution, a ELU nonlinearity and a batch normalization are applied to the stacked feature maps. The output has the same temporal resolution of the signal in input.

- The *segment classifier* module is designed to segment the high-frequency representation output of the decoder into the desired sleep stage prediction frequency. The module consist of a dense segmentation layer (*i.e.*, 1d convolution layer with a hyperbolic tangent activation function), an average-pooling layer (*e.g.*, with $kernel\_size = stride\_size = 30sec * f_s$ considering the same prediction frequency of a sleep scorer) and two 1D convolutional layers (the first using an ELU activation function, and the latter using a softmax activation function). The output of the segment classifier is a $L \times K$, where L is the number of segments and $K = 5$ is the number of sleep stages.

The sequence length L, the number of filters, the kernel and the stride sizes are specified in Figure 6.1. The softmax function, together with the cross-entropy loss function, is used to train the model to output the probabilities for the five mutually exclusive classes K that correspond to the five sleep stages. The architecture is trained end-to-end via backpropagation, using the sequence-to-sequence learning approach. The model is trained using mini-batch Adam gradient-based optimizer [91] with a learning rate lr. The training procedure runs up to a maximum number of iterations, as long as the break early stopping condition is satisfied.

6.1.2 Regularization Techniques

Unlike [57], we consider early stopping and data augmentation as regularization techniques. As stated in [94] "*regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error*". Early stopping and data augmentation do so in different ways, they both decrease the regularization error.

Early stopping. It provides guidance on how many iterations can be run before the model begins to overfit [93]. The training procedure is stopped as soon as the performance (*i.e.*, F1-score) on the validation set is lower than it was in the previous iteration step. In our experiments, before hastily stopping the learning procedure, the algorithm runs for an additional number of iterations (by fixing the so called *patience* parameter). The model with the highest performance is the one we finally save.

Data augmentation. The signals in input are randomly modified during training procedure to improve model generalization. Variable length of the sequences in input are replaced with a Gaussian noise. For each sample in a batch, with 0.1 probability, a fraction of the sequence is replaced with $N(\mu = \hat{\mu}, \sigma^2 = 0.01)$, where $\hat{\mu}$ is the mean of the sample's signals. The fraction is sampled with a log-uniform distribution $\{min = 0.001; max = 0.33\}$. With a 0.1 probability at most one channel is entirely replaced by noise.

6.1.3 Training Parameters

The training parameters (*e.g.*, adam-optimizer parameters beta1 and beta2, mini-batch size etc.) are all set as stated in [57]. The learning rate, the early stopping *patience* parameter and the maximum number of iterations have been changed to 10^-5, 100 and 1000 respectively, to let U-Sleep converge faster. The architecture has several hyperparameters (*e.g.*, number of layers, number/sizes of filters, regularization parameters, training parameters etc.) which could be optimized to tune its performance on any dataset. We decided to not systematically tune all these parameters, as this is out of our scope, but to fix them for all the experiments, as done in the original network.

6.2 Transfer Learning

We define Transfer Learning quoting the following clear and simple statements:

"*Transfer learning and domain adaptation refer to the situation where what has been learned in one setting* (e.g., *distribution* P_1*) is exploited to improve generalization in another setting (say, distribution* P_2*)*" by [94].

"*Given a source domain* D_S *and learning task* T_S*, a target domain* D_T *and learning task* T_T*, transfer learning aims to help improve the learning of the target predictive function* $f_T(\cdot)$ *in* D_T *using the knowledge in* D_S *and* T_S*, where* $D_S \neq D_T$ *and* $T_S \neq T_T$" by [113].

Formally, let $P_1 = D_\mathrm{S} = \{X_\mathrm{S}, Y_\mathrm{S}\}$ denote the data domain S with the biosignal/feature space X_S and the corresponding label space Y_S. Let T_S denote the task in the domain S maximizing the conditional probability distributions $P(y_\mathrm{S}|\mathbf{x}_\mathrm{S})$, where $\mathbf{x}_\mathrm{S} \in X_\mathrm{S}$ and $\mathbf{y}_\mathrm{S} \in Y_\mathrm{S}$. Similarly, let $P_2 = D_\mathrm{T} = \{X_\mathrm{T}, Y_\mathrm{T}\}$ denote the data domain T with the biosignal/feature space X_T and the corresponding label space Y_T. T_T denote the task in the domain T maximizing the conditional probability distributions $P(y_\mathrm{T}|\mathbf{x}_\mathrm{T})$, where $\mathbf{x}_\mathrm{T} \in X_\mathrm{T}$ and $\mathbf{y}_\mathrm{T} \in Y_\mathrm{T}$. The transfer learning technique aim to improve the learning of the distributions $P(y_\mathrm{T}|\mathbf{x}_\mathrm{T})$ given what we previously learned from D_S and T_S, where $D_\mathrm{S} \neq D_\mathrm{T}$ or $T_\mathrm{S} \neq T_\mathrm{T}$.

Overall the transfer learning approach have the following advantages: (1) The time-complexity of the training phase on the target domain (*i.e.*, fine-tuning procedure) is drastically reduced in respect to that required if the learning process is made from scratch; (2) it does not require a big-sized training set, making the approach feasible when labelled data are missing; (3) the source model is already pre-trained, hence the hyperparameters are already tuned/optimized. This allow to reach quickly high performances on the target task without re-designing the model and without lose its generalization capabilities. As highlighted in [113], we can define two macro-classes of transfer learning approaches. *Inductive* transfer refers to those scenario in which both the task and the domain are different in the source and target model; *transductive* transfer defines a method in which the model adapts to a different target domain where, however, the source and the target task are the same. *Transductive* transfer is exactly what we do in our experiments, where $T_\mathrm{S} \equiv T_\mathrm{T}$, as the task is always to perform sleep staging with the same set of sleep classes/stages.

The main challenges when handling with transfer learning are "*what*" transfer, "*when*" transfer and "*how*" transfer. "*What*" refers to the data domain shifts (*e.g.*,

different hardware, different subject distributions with different disorders) and the disagreement between the predictions of the model and labels given by different physicians. "*When*" is not the major issue here as we will perform the transfer only once, and directly on the whole target dataset available. "*How*" is the most challenging decision. The process involves overwriting a knowledge from a small-sized database to a previous big-sized knowledge (result of a long training process). In this scenario, one concern is to avoid ending up in what the data scientists call catastrophic forgetting:

"*Also known as catastrophic interference, it is the tendency of an artificial neural network to completely and abruptly forget previously learned information upon learning new information*" by [114].

Even if it is conceptually easy to understand, avoiding its occurrence is not trivial. To partially bypass this phenomena we fine-tune the architecture on the target domain using a smaller learning rate.

In our experiments we will first pre-train the architecture on the data-source domain S (*e.g.*, a set of different domains/databases $\{S_{\mathrm{DB}_1}, S_{\mathrm{DB}_2}, ..., S_{\mathrm{DB}_n}\}$), then we fine-tune (*i.e.*, re-train) the model on the data-target domain T. Formally, we first minimize the loss function L_S, resulting in the learned parameters θ:

$$argmin_\theta = \sum_{(\mathbf{x},\mathbf{y})\in D_S} L_S(\mathbf{x}, P(\mathbf{y}|\mathbf{x}), P_\theta(\mathbf{y},\mathbf{x})) \tag{6.1}$$

The parameters θ of the pre-trained model will be used as the starting point on the data-target domain T. To transfer the learning on the new domain T, we fine-tune all the pre-trained parameters $\theta' = \theta$ (*i.e.*, the entire network is further trained on the new data domain T):

$$argmin_{\theta'=\theta} = \sum_{(\mathbf{x},\mathbf{y})\in D_T} L_T(\mathbf{x}, P(\mathbf{y}|\mathbf{x}), P_\theta(\mathbf{y},\mathbf{x})) \tag{6.2}$$

6.3 Conditional Learning

Basically all the sleep scoring architectures learn in a conditional way. The aim is to maximize the conditional probability distributions $P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{X}$ are the sequences of the biosignals in input and $\mathbf{Y}$ are the corresponding ground-truth labels. For each epoch $\mathbf{x}_t$ in input the models aim to maximize the conditional probability distribution $P(y_t|\mathbf{x}_t)$, where y_t is the $t-th$ one-hot encoded vector of the ground-truth label. Hence, the model is trained to minimize the prediction error conditioned only by the knowledge of $\mathbf{X}$. We know that the sleep data $\mathbf{X}$ often come from different sources or data domains. Even in the same cohort, subjects with different demographics and sleep disorders may occur, resulting in significant shifts in their sleep data $\mathbf{X}$ distributions. Imagine to have in the same data cohort G different groups of subjects $\{g_1, g_2, ..., g_G\}$, with $g_1 = \{healthy\}$, $g_2 = \{sleep\_apnea\}$ and so on. This additional information about the group (*i.e.*, the sleep disorder group g_i) to which the subject belongs can be given in input to the model. So, we can either train G fully separated models, each maximizing G different $P(\mathbf{Y}|\mathbf{X})$ functions, or either train a single model maximizing the conditional probability distributions $P(\mathbf{Y}|\mathbf{X}, g_i)$. The latter - *i.e.*, train the joint model with the additional condition g_i - is the smartest

approach; the tasks are similar enough to benefit from sharing the parameters and the extracted features.

We decided to exploit the batch normalization layers to insert the additional knowledge in the training of our model. In literature different normalization variants have been proposed by modulating the parameters of the vanilla batch normalization (BN) layer [115]–[119]. We decided to exploit the Sandwich Batch Normalization (SaBN) approach recently proposed in [120].

The vanilla BN [86] normalizes the samples in a mini-batch in input by using the mean μ and the standard deviation σ, and then re-scales them with the γ and β parameters. So, given the feature in input $f \in \mathbb{R}^{B\times C\times H\times W}$, where B is the batch size, C is the number of channels and H and W are the height and width respectively, the vanilla BN computes:

$$h = \gamma(\frac{f-\mu(f)}{\sigma(f)}) + \beta \tag{6.3}$$

where $\mu(f)$ and $\sigma(f)$ are the mean and variance running estimates (batch statistics, *i.e.*, moving mean and moving variance) computed on f along (N, H, W) dimensions; γ and β are the re-scaling learnable parameters of the BN affine layer with shape C. Clearly, the vanilla BN has only a single re-scaling transform, indirectly assuming all features coming from a single data distribution. In [118], to tackle the data heterogeneity issue (*i.e.*, images from different data domains/distributions), they propose the Categorical Conditional BN (CCBN), so boosting the quality of the generated images. The CCBN layer computes the following operation:

$$h = \gamma_{\mathrm{g}}(\frac{f-\mu(f)}{\sigma(f)}) + \beta_{\mathrm{g}} \qquad g = 1, ..., G \tag{6.4}$$

where γ_{g} and β_{g} are the re-scaling learnable parameters of each $g-th$ affine layer, where g corresponds to the domain index associated to the input. The parameters of each affine layer are learned to capture the domain/distribution-specific information. In [120], instead, they propose the Sandwich Batch Normalization layer, an improved variant of the CCBN. They claim that different individual affine layers might cause an imbalanced learning for the different domains/distributions. They factorize the BN affine layer into one shared "sandwich" BN layer cascaded by a set of independent BN affine layers, computed as follows:

$$h = \gamma_{\mathrm{g}}(\gamma_{\mathrm{sa}}(\frac{f-\mu(f)}{\sigma(f)}) + \beta_{\mathrm{sa}}) + \beta_{\mathrm{g}} \qquad i = 1, ..., G \tag{6.5}$$

where γ_{sa} and β_{sa} are the re-scaling learnable parameters of the "sandwich" shared affine BN layer, while, as above, γ_{g} and β_{g} are the re-scaling learnable parameters of each $g-th$ affine layer, conditioned on the categorical input g. The SaBN enable the conditional fine-tuning of a pre-trained U-Sleep architecture, conditioned by the categorical index in input g.

6.4 Databases

We train and evaluate U-Sleep on 19578 recordings from 15322 subjects of 12 publicly available clinical studies, as done in [57]. Below a detailed description for each database:

ABC. The Apnea, Bariatric surgery, and CPAP database consists of 132 recordings from 49 patients with severe obstructive sleep apnea and morbity obesity (BMI from 35 to 45) [36], [37]. EEG signals (F4-M1, F3-M2, C4-M1, C3-M2, O2-M1, O1-M2) and EOG signals (E2-M1, E1-M2) have been considered in our experiments. The signals are recorded at 256Hz, and hardware low-pass filtered and high-pass filtered at 105Hz and at 0.16Hz respectively. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://doi.org/10.25822/nx52-bc11 and https://clinicaltrials.gov/ct2/show/NCT01187771.

CCSHS. The Cleveland Childrend's Sleep and Health Study consists of children and adolescents recordings. In our experiments we consider 515 recordings from adolescents aged 16-19 years. The recordings are collected in three different hospitals around Cleveland, Ohio, US [36], [38]. EEG signals (C4-A1, C3-A2) and EOG signals (ROC-A1, LOC-A2) have been considered in our experiments. The signals are recorded at 128Hz, and hardware high-pass filtered at 0.15Hz. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://doi.org/10.25822/cg2n-4y91.

CFS. The Cleveland Family Study is a family-based study on sleep apnea disordered subjects. The database consists of 2284 subjects from 361 families [36], [39]. We consider recordings of 730 subjects from 144 families (whence full whole-night PSG were available). For this specific database, the data split (train/val/test set) is done by considering subjects and family belonging (*i.e.*, all the family members appear in the same data split). EEG signals (C4-A1, C3-A2) and EOG signals (ROC-A1, LOC-A2) have been considered in our experiments. The signals are recorded at 128Hz, and hardware low-pass filtered and high-pass filtered at 105Hz and 0.16 Hz respectively. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://doi.org/10.25822/jmyx-mz90.

CHAT. The Childhood Adenotonsillectomy Trial database consists of 1638 recordings (452 baseline, 407 follow-up and 779 control) from 1232 children post-adenotonsillectomy-surgery aged 5-10 years. The recordings are collected in six different sleep centers in Massachusetts, Missouri, New York, Ohio and Pennsylvania [36], [40], [41]. EEG signals (F4-M1, F3-M2, C4-M1, C3-M2, O2-M1, O1-M2, T4-M1, T3-M2) and EOG signals (E2-M1, E1-M2) have been considered in our experiments. The signals are recorded at 200Hz (or higher in other sleep centers), and different hardware filtering given the different acquisition systems. One recording has been excluded - EOG missing. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://doi.org/10.25822/d68d-8g03 and https://clinicaltrials.gov/ct2/show/NCT00560859.

DCSM. The Danish Centre for Sleep Medicine database consists of 255 recordings from patients with potential and non-specific sleep related disorders. No demographic is available for the database. EEG signals (F4-M1, F3-M2, C4-M1,

C3-M2, O2-M1, O1-M2, T4-M1, T3-M2) and EOG signals (E2-M1, E1-M2) have been considered in our experiments. The signals are recorded at 256Hz, and band-pass filtered between 0.3Hz and 70Hz. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://sid.erda.dk/wsgi-bin/ls.py?share_id=fUH3xbOXv8.

HPAP. The Home Positive Airway Pressure database consists of 373 recordings (238 considered in our experiments) from obstructive sleep apnea patients aged over 18 years. The recordings are collected in seven different US sleep centers [36], [44]. EEG signals (F4-M1, F3-M2, C4-M1, C3-M2, O2-M1, O1-M2, T4-M1, T3-M2) and EOG signals (E2-M1, E1-M2) have been considered in our experiments. The signals are recorded at 200Hz, no filtering applied. Nine recordings have been excluded - EOG and/or reference channels missing. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://doi.org/10.25822/xmwv-yz91 and https://clinicaltrials.gov/ct2/show/NCT00642486.

MESA. The Multi-Ethnic Study of Atherosclerosis consists of 2237 recordings (2056 considered in our experiments) from a cohort of black, white, Hispanic and Chinese-American subjects aged 45-84 years [36], [45]. EEG signals (Fz-Cz, C4-M1, Cz-Oz) and EOG signals (E2-Fpz, E1-Fpz) have been considered in our experiments. The signals are recorded at 256Hz, and hardware low-pass filtered at 100Hz. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://doi.org/10.25822/n7hq-c406.

MROS. The database is a subset of the larger study Osteoporotic Fractures in Men (MrOS), involving 5,994 community-dwelling men aged over 65 years [36], [46], [47]. In our experiments we consider 3926 recordings (2900 from visit 1 and 1026 from visit 2) from 2903 subjects, which underwent in-home overnight PSG. EEG signals (C4-A1, C3-A2) and EOG signals (ROC-A1, LOC-A2) have been considered in our experiments. The signals are recorded at 256Hz, and hardware high-pass filtered at 0.15Hz. Seven recordings have been excluded - EOG channels and/or sleep stage annotation files missing. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://doi.org/10.25822/kc27-0425.

PHYS. The database from the 1028 PhysioNet/CniC Challenge consists of 1985 recordings (994 labelled considered in our experiments) from patients with potential sleep disorders [48], [49]. EEG signals (F4-M1, F3-M2, C4-M1, C3-M2, O2-M1, O1-M2) and one EOG signal (E1-M2) have been considered in our experiments. The signals are recorded at 200Hz. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://physionet.org/content/challenge-2018/1.0.0/.

SEDF-SC & SEDF-ST. The Sleep-EDF Expanded database consists of 197 recordings from two subset studies. The Sleep-EDF Sleep Cassette (SEDF-SC) consists of 153 recordings from 78 healthy subjects aged 25-101 years. The Sleep-EDF Sleep Telemetry (SEDF-ST) consists of 44 recordings from 22 healthy subjects with mild difficulty falling asleep (two recordings collected for each subject, *i.e.*, one after

temazepam intake and one after placebo intake) [32], [48]. EEG signals (Fpz-Cz, Pz-Oz) and one EOG signal (ROC-LOC) have been considered in our experiments. The signals are recorded at 100Hz. The recordings are manually scored by sleep experts according to the R&K scoring rules, and re-aligned to the AASM rules as described at the end of this section. For more information we refer to https://doi.org/10.13026/C2C30J.

SHHS. The Sleep Heart Health Study consists of 8444 recordings (5793 from visit 1 and 2651 from visit 2) from 5797 patients with sleep-disordered breathing aged over 40 years [34], [36]. EEG signals (C4-A1, C3-A2) and EOG signals (ROC-A1, LOC-A2) have been considered in our experiments. The EEG and EOG signals are recorded at 125Hz and 50Hz respectively, and hardware high-pass filtered at 0.15Hz. The recordings are manually scored by sleep experts according to the R&K scoring rules, and re-aligned to the AASM rules as described at the end of this section. For more information we refer to https://clinicaltrials.gov/ct2/show/NCT00005275 and https://doi.org/10.25822/ghy8-ks59.

SOF. The database is a subset of the larger study Osteoporotic Fractures (SOF). In our experiments we consider 453 recordings (from visit 8), which underwent in-home overnight PSG [36], [50], [51]. EEG signals (C4-A1, C3-A2) and EOG signals (ROC-A1, LOC-A2) have been considered in our experiments. The EEG and EOG signals are recorded at 128Hz, and hardware high-pass filtered at 0.15Hz. The recordings are manually scored by sleep experts according to the R&K scoring rules, and re-aligned to the AASM rules as described at the end of this section. For more information we refer to https://doi.org/10.25822/e1cf-rx65.

In our experiments we also exploit the Bern Sleep Data Base (*BSDB*) registry, the sleep disorder patient cohort of the Inselspital, University hospital Bern. The recordings have been collected from 2000 to 2021 at the Department of Neurology, at the University hospital Bern. Secondary usage was ethically approved (KEK-Nr. 2020-01094). The dataset consists of 8950 recordings from healthy subjects and patients aged 0-91 years. The strength of this dataset is that, unlike the ones available online, it contains patients covering the full spectrum of sleep disorders, many of whom were diagnosed with multiple sleep disorders and non-sleep related comorbidities [121]; thus providing an exceptionally heterogeneous PSG data set. EEG (F4-M1, F3-M2, C4-M1, C3-M2, O2-M1, O1-M2) and EOG (E2-M1, E1-M2) standard derivations have been considered in our experiments. The signals are recorded at 200Hz. The recordings are manually scored by sleep experts according to the AASM rules.

An overview of all the open access (OA) datasets and the *BSDB* dataset along with demographic statistics is reported in Table 6.1.

The data pre-processing and data selection/sampling across all the datasets is the same as in [57].

Data pre-processing. The signals are resampled to 128Hz and rescaled (per channel and per-subject), so that, for each channel, the EEG signal has median 0 and inter quartile range (IRQ) 1. The values with an absolute deviation from the median above 20*IQR are clipped. The signals outside the range of the scored hypnogram are trimmed. The recordings scored according to R&K rules results in six scoring

TABLE 6.1: **Datasets overview with demographic statistics.** Missing values are due to study design or anonymized data. On the *BSDB* dataset, we compute the age and the sex values on the 99.1% and on the 98.6% of the whole dataset, respectively, because of missing age/sex information. Datasets directly available online are identified by ✓, whilst datasets that require approval from a Data Access Committee marked by (✓). *BSDB* is a private dataset.

	Datasets	Recordings	Age in years ($\mu \pm \sigma$)	Sex % (F/M)
[36], [37]	ABC (✓)	132	48.8 ± 9.8	43/57
[36], [38]	CCSHS (✓)	515	17.7 ± 0.4	50/50
[36], [39]	CFS (✓)	730	41.7 ± 20.0	55/45
[36], [40], [41]	CHAT (✓)	1638	6.6 ± 1.4	52/48
-	DCSM ✓	255	-	-
[36], [44]	HPAP (✓)	238	46.5 ± 11.9	43/57
[36], [45]	MESA (✓)	2056	69.4 ± 9.1	54/46
[36], [46], [47]	MROS (✓)	3926	76.4 ± 5.5	0/100
[48], [49]	PHYS ✓	994	55.2 ± 14.3	33/67
[32], [48]	SEDF-SC ✓	153	58.8 ± 22.0	53/47
[32], [48]	SEDF-ST ✓	44	40.2 ± 17.7	68/32
[34], [36]	SHHS (✓)	8444	63.1 ± 11.2	52/48
[36], [50], [51]	SOF (✓)	453	82.8 ± 3.1	100/0
-	*BSDB*	8884	47.9 ± 18.4	66/34

classes, *i.e.*, awake, N1, N2, N3, N4, and REM. In order to use the AASM standard, we merge the N3 and N4 stages into a single stage N3. The loss function for stages as MOVEMENT and UNKNOWN is masked during the training procedure.

Data sampling. U-Sleep is trained using mini-batch Adam gradient-based optimizer. Each element in the batch is a sequence/segment of $L = 35$ EEG and EOG 30-second signals/epochs from a single subject. Each sequence/element is sampled from the training data as follows. (1) dataset sampling: one dataset is selected randomly. The probability that a dataset D is selected is given by $P(D) = \alpha P_1(D) \cdot (1 - \alpha) P_2(D)$, where $P_1(D)$ is the probability that a dataset is sampled with a uniform distribution $1/N_\mathrm{D}$, where N_D is the number of available datasets, and $P_2(D)$ is the probability of sampling a dataset according to its size. The parameter α was set to 0.5 to equally weight $P_1(D)$ and $P_2(D)$; (2) subject sampling: a recording S_D is uniformly sampled from D; (3) channel sampling: one EEG and one EOG are uniformly sampled from the available combinations of channels in S_D (*e.g.*, if 2 EEG and 2 EOG channels are available, four combinations would be possible); (4) segment sampling: a segment of EEG/EOG signals of length $L = 35$ is selected as follows: first uniformly sample a class from $W, N1, N2, N3, R$, then select randomly a 30-second epoch scored with the sampled class and finally shift the chosen epoch in a random position of the segment of length L.

6.5 Results

6.5.1 Experiment Designs

Unlike the recommendation of the AASM manual, during the pre-processing procedure no filtering was applied to the EEG and the EOG signals. Most importantly, we found that in their original model implementation the data extraction, and the resulting sampling procedure, were extracting atypical channel derivations, see Supplementary Table B.5, interestingly different to those recommended in the AASM guidelines. In this study, we want to test the resilience of U-Sleep to the strict AASM guidelines. To this aim, we extract the channel derivations following the guidelines (as meant to be done in [57]), to better understand the impact of channel selection on the overall performance.

Below we summarize all the experiments performed on U-Sleep:

(i) We pre-train U-Sleep on all the OA datasets using both the original implementation selecting the atypical channel derivations (*U-Sleep-v0*), and our adaptation following AASM guidelines (*U-Sleep-v1*). We split each dataset in training (75%), validation (up to 10%, at most 50 subjects) and test set (up to 15%, at most 100 subjects). The split of the PSG recordings is done per-subject or per-family, *i.e.*, recordings from the same subject or members of the same family appear in the same data split. In Table 6.2 we summarize the data split on each open access dataset. We evaluate both *U-Sleep-v0* and *U-Sleep-v1* on the test set of the *BSDB* dataset. We also evaluate the models on the whole $BSDB_{(100\%)}$ dataset, to test on a higher number of subjects, with a higher heterogeneity of sleep disorders and a wider age range. A model pre-trained on the open access datasets and evaluated directly on the *BSDB* dataset is what we will refer to as direct transfer (DT) on *BSDB*.

(ii) We exploit the *BSDB* dataset to evaluate whether a DL based scoring architecture, trained with a large and a highly heterogeneous database, is able to generalize on the open access datasets from different data centers. We split the *BSDB* recordings in training (75%), validation (10%) and test set (15%). We run two different experiments on *U-Sleep-v1*: we train the model from scratch (S) on the *BSDB* dataset; we fine-tune (FT) the model pre-trained in *(i)* on the *BSDB* dataset, by using the transfer learning approach (see subsection 6.2). Then, we evaluate both (S) and (FT) on the test set of all the OA datasets and the test set of the *BSDB* dataset.

(iii) We exploit the *BSDB* dataset to investigate whether U-Sleep needs to be trained by also having access to chronological age-related information. We split the *BSDB* dataset in seven groups, according to the age categories of the subjects [105], resulting in $G = 7$ subdatasets, see Supplementary Analysis section A.1. We further split the recordings of each subdataset in training (75%), validation (10% at most 50 subjects) and test set (15% at most 100 subjects). We run three different experiments on *U-Sleep-v1*: we fine-tune the model by using all the training set of the seven groups (FT); we fine-tune seven Independent models by using the training set of each group independently (FT-I); we fine-tune a single Sandwich Batch Normalization model (exploiting the batch normalization layers, see subsection 6.3), to add the condition on the age-group-index G for each recording (FT-SaBN). These last two experiments have been replicated considering only two age groups, *i.e.*,

TABLE 6.2: **Data split on the open access (OA) datasets.** We report the total number of recordings, and the number of recordings used to train, validate and test the U-Sleep architecture in the experiment (i).

Datasets	Recordings	Train	Valid	Test
ABC	132	93	15	24
CCSHS	515	387	50	78
CFS	730	531	95	104
CHAT	1638	1438	70	130
DCSM	255	190	26	39
HPAP	238	178	24	36
MESA	2056	1906	50	100
MROS	3926	3728	69	129
PHYS	994	844	50	100
SEDF-SC	153	115	15	23
SEDF-ST	44	30	6	8
SHHS	8444	8226	77	141
SOF	453	339	46	68

TABLE 6.3: **Data split on the BSDB dataset.** We report the total number of recordings, and the number of recordings used to train, validate and test the U-Sleep architecture in the experiments (ii) and (iii).

Datasets		Recordings	Train	Valid	Test
(ii)	-	8884	6658	882	1344
(iii)	B	151	111	14	26
	C	246	185	26	35
	A	177	132	17	28
	YA	2066	1902	58	106
	MA	3636	3482	51	103
	E	1539	1378	55	106
	OE	988	829	53	106

babies/children and adults, as recommended in [2], resulting in two additional fine-tuned model (FT-I and FT-SaBN for $G = 2$). We evaluate all the fine-tuned models on the independent test test of each age group.

In Table 6.3 we summarize the two different data split, in experiment *(ii)* and experiment *(iii)*, on the *BSDB* dataset.

6.5.2 Metrics

In all our experiments we evaluate U-Sleep as stated in [57]. The model scores the full PSG, without considering the predicted class on a segment with a label different

TABLE 6.4: **Experiments *(i)*: *U-Sleep-v0* and *U-Sleep-v1* F1-score.** *(i)* Performance of *U-Sleep-v0* and *U-Sleep-v1*, pre-trained on the open access (OA) datasets, and evaluated on the test set of the *BSDB* dataset, and on the whole $BSDB_{(100\%)}$ dataset, *i.e.*, both direct transfer (DT) on *BSDB*. We report the F1-score (%F1), specifically the mean value and the standard deviation ($\mu \pm \sigma$) computed across the recordings.

	Training on OA	
Datasets	*U-Sleep-v0*	*U-Sleep-v1*
BSDB	72.5 ± 12.2	72.5 ± 12.0
$BSDB_{(100\%)}$	72.9 ± 12.4	72.9 ± 12.4

from the five sleep stages (*e.g.*, segment labelled as 'UNKNOWN' or as 'MOVEMENT'). The final prediction is the results of all the possible combinations of the available EEG and EOG channels for each PSG. Hence, we use the majority vote, *i.e.*, the ensemble of predictions given by the multiple combination of channels in input.

The unweighted F1-score metric [89] has been computed on all the testing sets to evaluate the performance of the model on all the experiments. We compute the F1-score for all the five classes, we then combine them by calculating the unweighted mean. Note that the unweighted F1-scores reduce the absolute scores due to lower performance on less abundant classes such as sleep stage N1. For this reason, we also report in Supplementary Tables B.6, B.7, B.8 and B.9 the results achieved in terms of weighted F1-score - *i.e.*, the metric is weighted by the number of true instances for each label, so as to consider the high imbalance between the sleep stages. In that case, the absolute scores significantly increases on all the experiments.

6.5.3 Analysis of Experiments

(i) *Clinically non-recommended channel derivations*. In Table 6.4 we compare the performance of U-Sleep pre-trained on all the OA datasets, with (*U-Sleep-v0*) and without (*U-Sleep-v1*) using randomly ordered channel derivations. There is no statistically significant difference between the two differently trained architectures evaluated on the test set of the *BSDB* dataset (paired t-test $p - value > 0.05$). Most importantly, we found no difference in performance with the direct transfer also on the whole $BSDB_{(100\%)}$ dataset (paired t-test $p - value > 0.05$). These results clearly show how the architecture is able to generalize regardless of the channel derivations used during the training procedure, also on a never seen highly heterogeneous dataset.

(ii) *Generalizability on different data centers with a heterogeneous dataset*. In Table 6.5 we report the results obtained on *U-Sleep-v1* pre-trained in *(i)* on the open access (OA) datasets, and evaluated on all the test sets of the open access datasets and on the test set of the *BSDB* dataset. We also show the results obtained on *U-Sleep-v1* trained from scratch (S) on the *BSDB* dataset, and the results obtained on the model pre-trained in *(i)* on OA and then fine-tuned (FT) on the *BSDB* dataset. Unlike what we expected, both the models (S) and (FT), trained with a large and a highly heterogeneous database, are

TABLE 6.5: **Experiments *(ii)*: *U-Sleep-v1* F1-score.**
(ii) Performance of *U-Sleep-v1*, pre-trained on the open access (OA) datasets, and evaluated on all the test sets of the open access datasets and on the test set of the *BSDB* dataset. We also report the performance of *U-Sleep-v1* trained from scratch (S) or fine-tuned (FT) on the *BSDB* dataset, and evaluated on all the test sets of all the available datasets. We report the F1-score (%F1), specifically the mean value and the standard deviation ($\mu \pm \sigma$) computed across the recordings.

Datasets	Training on OA	Training on *BSDB*	
	U-Sleep-v1	*U-Sleep-v1* (S)	*U-Sleep-v1* (FT)
ABC	73.6 ± 11.4	71.4 ± 13.9	69.0 ± 12.5
CCSHS	84.9 ± 5.1	77.3 ± 7.2	77.3 ± 6.7
CFS	76.6 ± 11.6	70.2 ± 10.8	70.9 ± 10.2
CHAT	82.1 ± 6.5	72.9 ± 8.0	68.8 ± 8.7
DCSM	79.3 ± 9.3	71.5 ± 11.2	69.3 ± 10.5
HPAP	73.8 ± 10.8	68.9 ± 11.1	67.9 ± 12.5
MESA	72.7 ± 10.8	68.5 ± 14.3	68.7 ± 11.9
MROS	71.4 ± 12.1	61.7 ± 13.7	63.9 ± 13.2
PHYS	74.2 ± 10.7	72.9 ± 11.2	73.2 ± 11.4
SEDF-SC	77.8 ± 7.9	75.8 ± 8.0	77.9 ± 7.7
SEDF-ST	77.2 ± 10.1	64.3 ± 15.4	67.5 ± 12.4
SHHS	76.9 ± 9.7	70.9 ± 9.3	73.0 ± 8.9
SOF	74.8 ± 9.8	64.6 ± 12.6	67.5 ± 11.2
avg OA	76.5 ± 10.6	69.9 ± 11.9	70.2 ± 11.1
BSDB	72.5 ± 12.0 (DT)	77.6 ± 11.3	77.3 ± 11.4

not able to generalize on the open access datasets from the different data centers. The average performance achieved on the OA with (S) and (FT) models is significantly lower compared to the performance of the model pre-trained on OA (paired t-tests $p - value < 0.001$). Whilst, with both (S) and (FT) we show a significant increase in performance compared to the direct transfer (DT), on the test set of the *BSDB* dataset (paired t-tests $p - value < 0.001$). We also found that the training from scratch results in significantly higher performance (paired t-test $p - value < 0.001$) on the *BSDB* dataset, compared to the performance of the fine-tuned model. No significant difference (paired t-test $p - value > 0.05$) occurs between (S) and (FT) evaluated on the average performance on OA. The pre-training on the OA dataset it is not beneficial on the *BSDB* dataset. With a high number of highly heterogeneous subjects we can directly train the model from scratch on the dataset. However, we have to mention that the fine-tuned model reach the the local optimum in a fewer number of iterations (number of iterations: FT $= 382 <$ S $= 533$).

TABLE 6.6: **Experiments *(iii)*: *U-Sleep-v1* F1-score.**
(iii) Performance of *U-Sleep-v1* on a single model fine-tuned on all the training set of the seven *BSDB* groups (FT); on seven/two models fine-tuned on the independent training set of each group (FT-I) with $G = 7$ and $G = 2$ respectively; and on a single model fine-tuned on all the training set of the seven/two *BSDB* groups conditioned (FT-SaBN) by $G = 7$ and by $G = 2$ groups respectively. All the fine-tuned models are evaluated on the associated test set of each group. We report the F1-score (%F1), specifically the mean value and the standard deviation $(\mu \pm \sigma)$ computed across the recordings. B: Babies (0-3 years); C: Children (4-12 years); A: Adolescents (13-18 years); YA: Young Adults (19-39 years); MA: Middle Aged Adults (40-59 years); E: Elderly (60-69 years); OE: Old Elderly (> 70 years). When $G = 2$ we have the following two groups $G_1 = \{B \cup C\}$, $G_2 = \{A \cup YA \cup MA \cup E \cup OE\}$, further details in Supplementary Analysis section A.1

Subsets	FT	FT-I		FT-SaBN	
	(G=1)	(G = 7)	(G = 2)	(G = 7)	(G = 2)
B	74.9 ± 6.8	74.1 ± 6.6 ^{G_1}	74.8 ± 6.2 ^{G_1}	72.2 ± 7.7	72.6 ± 7.7
C	75.0 ± 9.8	74.9 ± 9.2 ^{G_2}	75.9 ± 9.1 ^{G_1}	74.8 ± 8.9	75.6 ± 10.1
A	82.7 ± 13.7	80.0 ± 14.6 ^{G_3}	82.8 ± 13.6 ^{G_2}	82.3 ± 13.7	82.0 ± 14.0
YA	80.8 ± 11.5	80.6 ± 11.6 ^{G_4}	80.6 ± 11.6 ^{G_2}	80.3 ± 11.9	79.9 ± 11.9
MA	80.4 ± 7.8	79.90 ± 8.0 ^{G_5}	79.8 ± 8.2 ^{G_2}	79.6 ± 8.0	79.4 ± 8.3
E	75.7 ± 10.1	74.2 ± 10.7 ^{G_6}	74.9 ± 10.2 ^{G_2}	74.5 ± 10.6	73.9 ± 10.9
OE	75.2 ± 11.7	73.9 ± 11.0 ^{G_7}	74.9 ± 11.3 ^{G_2}	73.8 ± 11.7	74.0 ± 11.3
avg	77.9 ± 10.7	77.0 ± 10.8	77.6 ± 10.7	76.9 ± 11.0	76.8 ± 11.1

(iii) Training conditioned by age. In Table 6.6 we show the performance of *U-Sleep-v1* fine-tuned (FT) on all the training set of the seven *BSDB* groups, *i.e.*, single model. We also report the performance achieved using the training set of each group independently (FT-I) with $G = 7$ and $G = 2$ respectively (*i.e.*, seven and two models), and the performance achieved using the training set of the seven/two *BSDB* groups conditioned (FT-SaBN) by $G = 7$ and by $G = 2$ groups respectively (*i.e.*, single model). The mean and the standard deviation of the F1-score (%F1), are computed across the recordings of the test set of each of the seven *BSDB* age groups. Comparing both the experiments (FT-I and FT-SaBN) and types of grouping (G=2 and G=7) with the baseline (FT), we did not find a statistically significant increase of the performance in any of the subgroups (paired t-test $p - value > 0.05$). Despite the lack of significant performance differences in our age-conditioned models, REM sleep seems to be less accurately predicted for small children, if the training data set only consists of data from adults (see Supplementary Figure B.22, confusion matrix for test $\{CH\}$ against Model 1b). This is an interesting finding since small children exhibit more REM sleep (see Supplementary Figure B.20). Visual scoring guidelines for small children differ from the guidelines for adults, with REM sleep scoring strongly relying on irregular respiration [122]. However, overall these results show that, despite the age-related differences, the architecture

TABLE 6.7: ***U-Sleep-v1* F1-score and uncertainty estimate.** Performance of *U-Sleep-v1*, pre-trained on the open access (OA) datasets, and evaluated on all the test sets of the open access datasets (*avg* OA) with and without (*i.e.*, *U-Sleep-v1* pre-trained in *(ii)*) label smoothing. We report the F1-score (%F1), referred to the epochs kept after the $\mu_{\max}$ query selection procedure (q% threshold value fixed to 5%), and we report percentage of misclassified epochs among the rejected with query (%miscl.). Specifically, we report the mean value and the standard deviation ($\mu \pm \sigma$) computed across the recordings.

Dataset		*w/o* label smoothing	*w/* label smoothing
avg OA	%F1	78.1 ± 11.2	72.5 ± 12.4
	%miscl.	51.1 ± 9.5	53.7 ± 10.6

is able to deal with different age subgroups at the same time, without needing to have access to chronological age-related information during the training procedure.

6.5.4 Uncertainty estimate

The predicted sleep stage for each fixed-length $i > 0$ window comes with a probability value $\hat{p}$. As explained in the previous chapters, the probability value associated with the predicted sleep stage should mirror its ground truth correctness likelihood. When this happens the model is well calibrated. We also learned that label smoothing [95] has been shown to be a suitable technique to improve the calibration of the model. In our experiments, we also train U-Sleep with the label smoothing techniques, to add some noise on the labels, to better calibrate the model and to evaluate its impact on our uncertainty estimate procedure. When the model is trained with the label smoothing technique, the hard targets are smoothed with the standard uniform distribution $1/K$ (eq. 5.3), where K is the number of sleep stages. We fix the α smoothing parameter equal to 0.1.

We exploit the ensemble of the M different predictions (i.e. one prediction for each combination of channel in input) and the query procedure introduced in Chapter 4 to estimate the model uncertainty, and consequently the uncertain predictions. We can compute the mean $\mu_{\mathrm{i,k}}$ and the variance ${\sigma^2}_{\mathrm{i,k}}$ of the M predictions for each sleep stage k. The final prediction $\hat{y}_{\mathrm{i}}$ of the model will be given by $max(\mu_{\mathrm{i}})$, which we will refer to as $\mu_{\max}$, along with the assigned variance value ${\sigma^2}_{\mu\max}$. The mean $\mu_{\max}$ and the variance ${\sigma^2}_{\mu\max}$ can be used as indicators of the model uncertainty. High $\mu_{\max}$ and low ${\sigma^2}_{\mu\max}$ indicate that the model is confident about its prediction, i.e. low degree of uncertainty. In this chapter, we use only the $\mu_{\max}$ query procedure, *i.e.*, on each subject we select a fixed percentage of epochs with the lowest $\mu_{\max}$ value. It has been shown to be more efficient compared to the query procedure via ${\sigma^2}_{\mu\max}$. The query procedure requires the selection of a threshold q% on the distribution of the mean values. The selection criterion of the threshold value q% is based on a reasonable percentage of epochs to be re-sent to the physician for a secondary review. We fix q% equal to 5%, as done in the previous chapters.

We found that training *U-Sleep-v1* on the OA datasets with the label smoothing technique results in a significant decrease in performance compared to the baseline pre-trained in *(ii)* without label smoothing (paired t-test $p - value < 0.001$). Nonetheless, by adding some noise on the label during the training, we are able to select a significantly higher number of misclassified epochs among the selected ones (paired t-test $p - value < 0.001$). We thus succeed to better detect the uncertain predictions, see Table 6.7.

6.6 Discussion

In this Chapter we demonstrate how resilient to sleep complexity is U-Sleep, a DL based scoring architecture. We focused on three of the most significant aspects: channel derivation selection, multi centre heterogeneity needs and age conditioned fine tuning. Channel derivations do have complementary information, and a DL based model resulted resilient enough to be able to extract sleep patterns also from mismatched, atypical and clinically non-recommended derivations. We showed that the variability among different sleep data centers (*e.g.*, hardware, scoring rules etc.) needs to be taken into account more than the variability inside one single sleep center. A large database such as the *BSDB* (sleep disorder patient cohort of the Inselspital, with patients covering the full spectrum of sleep disorders) did not have enough heterogeneity to strengthen the performance of the DL based model on unseen data centers. Lastly, we show that a state-of-the-art DL network is able to deal with different age groups simultaneously, mitigating the need of adding chronological age-related information during training.

To our knowledge, our study on the automatic sleep scoring task is the largest in terms of number of polysomnography recordings and diversity with respect to both patient clinical pathology and age spectrum.

Chapter 7

Conclusions

In this thesis, we deeply investigated the resilience of the DL based scoring algorithms in solving the sleep scoring tasks. Our aims were the following: to increase the performance of existing sleep scoring algorithms, whilst quantifying the disagreement between their final sleep stages predictions and the annotations given by the physicians; to study the ability of these architectures in encoding the high inter-scorer variability and the high data variability from different sleep labs. The primary step toward these goals was to simplify an existing state-of-the-art sleep scoring architecture, while maintaining its performance. In Chapter 3 we first proposed to tackle the sleep scoring tasks by using simple feedforward based architectures, achieving comparable results to those using recurrent layers. In Chapter 4 we introduced, for the first time in sleep scoring, a novel approach to better calibrate the model and to further enhance the performance of the sleep scoring architecture. We exploited ensemble learning based algorithms together with label smoothing techniques. We also introduced an uncertainty estimate procedure to identify the most challenging sleep stage predictions, so as to quantify the disagreement between the algorithm and the physicians. All along the way, we showed the efficiency of these methodologies on different scoring architectures and sleep databases. In Chapter 5 we proposed to use the label smoothing technique along with the soft-consensus distribution in the training procedure of our model. The approach enabled the sleep scoring model to better adapt to the consensus of the multiple-scorers. Hence, we clearly demonstrate that a scoring algorithm is able to learn the inter-scorer variability. Finally, in the last Chapter 6 we came with two important findings: a DL based architecture does not need to be trained following the strict AASM guidelines, *e.g.* it solves the scoring task even by using clinically non-conventional channel derivations, with no need to receive in input additional information about the chronological age of the subjects; using data from multiple data centers always results in a better performing model compared with training on a single data cohort.

The above findings leave room to the following additional observations, open questions and possible directions as a continuation of this work.

Learning from multi-scored databases. The possibility of exploiting the full set of information that is hidden in a multi-scored dataset would certainly enhance automated DL algorithms performance. However, in order to generalize the approach proposed in Chapter 5, there are two big limitations. The first is that a far bigger datasets, highly heterogeneous (with different diagnosis, age range, gender etc.) scored by multiple physicians would be necessary. The second is that to transfer the consensus variability from a dataset to another would require finding a relation between the consensus variability and the complex, not easy to define, DL extracted

features related to the epoch itself.

Need to rebuild the AASM scoring rules. AASM scoring rules have been widely criticized over the years, for many aspects. The scoring manual has been designed to consider the sleep stages almost as discrete entities. However, it is well-known that sleep should be viewed as a continuum/gradual transition from one stage to another. A growing consensus suggests that we should reconsider the scoring rules and the entire scoring procedure. Given the high variability among the individual scorers and different sleep centers, more efforts should be made to improve the standardization of the methodology. This variability inevitably affects the performance of any kind of algorithm, since all algorithms are learning from the noisy variability of labels given by physicians. A relevant finding of this thesis is indeed that the heterogeneity given by data coming from different sleep data centers (*e.g.*, different sleep scorers) is much more relevant than the variability coming from patients affected by different sleep disorders. This latter insight raises a research question yet to be answered, *i.e.*, how could we define and quantify the heterogeneity of a sleep database? To what extent could we consider a database heterogeneous enough, to allow the algorithm to generalize across different data domains/centers?

Encoding clinically relevant sleep patterns from non-conventional channel derivations. The resilience of a DL based model to the atypical or non-conventional channel derivations is fascinating. The model still learns relevant sleep patterns while solving the scoring tasks with high state-of-the-art performance on multiple large-scale-heterogeneous data cohorts. Although this is a remarkable finding, it would be useful to further investigate the reasons why the DL model is still able to encode clinically valid information. DL has been criticised for its non-interpretability and its black-box behavior, factors that may actually limit its implementation in sleep centers. Future works should focus on solving the following open question: which sleep patterns/features or which brain regions our DL algorithms are encoding/highlighting from the typical/atypical channel derivations?

Encoding biomarkers of consciousness under conditions of abnormal cortical dynamics. As a follow-up study of Chapter 6, in section A.2 we demonstrated that a DL architecture, if properly pre-trained, it is actually also able to recognize consciousness (awake) and unconsciousness (NREM) states under conditions of abnormal cortical dynamics, by only looking at the EEG activity. Specifically, the algorithm pre-trained on a huge and heterogeneous dataset (raw data from healthy subjects and patients with different sleep disorders) is able to generalize on a dataset of subjects with genetic disorders, never seen by the algorithm, and characterized by abnormal sleep physiology. The algorithm is remarkably learning, from the raw EEG data, general patterns (noise/artifacts helps to discern awake and NREM states), still useful for the consciousness related task. Can we extract from the hidden knowledge of the layer of our scoring architecture meaningful biomarkers/features related to the conscious and unconscious states of these patients?

Our final consideration relates to the general approach to solve the automated scoring dilemma. DL algorithms have reached better performance than feature based approach, DL is definitely able in learning features alone. DL better resemble the unconscious wide and comprehensive knowledge of the human beings, that cannot be discretized in a set of well defined features. Being the AASM so widely criticized, being the sleep labels so noisy (*e.g.*, due to high inter- and intra- scorer variability), and being sleep so complex: could we delegate to a DL algorithm totally the scoring procedure? Could an unsupervised approach, that does not use labels, be the solution?

Appendix A

Supplementary Analysis

A.1 U-Sleep: Age analysis on *BSDB*

(G=7) Age Groups by [105]

Inspired by the meta-analysis of quantitative sleep parameters from childhood to old age reported in [105], we decided to study, and then run all our experiments, on the following seven age groups: Babies (B; 0-3 years), Children (C; 4-12 years), Adolescents (A; 13-18 years), Young Adults (YA; 19-39 years), Middle Aged Adults (MA; 40-59 years), Elderly (E; 60-69 years), Old Elderly (OE; >70 years). Unlike in [105], we also considered the additional group of Babies, uncovered in their study. In Supplementary Figure B.10 and in Table A.1, respectively, we show the age distribution of the *BSDB* dataset in the seven age groups, and we report the number of recordings, the age range, the age mean and standard deviation ($\mu \pm \sigma$) and the male/female percentage (M/F) for each group. For each PSG of the *BSDB* dataset we compute the following ten sleep parameters: Total Sleep Time (TST), Sleep Period Time (SPT), Wake After Sleep Onset (WASO), Sleep Latency (SL), Sleep Efficiency (SE), Percentage of N1 stage (pN1), Percentage of N2 stage (pN2), Percentage of N3 stage (pN3), Percentage of REM stage (pREM) and Number of stage shifts per hour (n_shift). In Table A.2 we report the mean and standard deviation ($\mu \pm \sigma$) of each sleep parameter for each age group. We also show the boxplots in Supplementary Figures from B.12 to B.21 computed on each sleep parameter and for each age group. In some of these plots emerge the continuous positive/negative trend on the specific sleep parameter from babies to old elderly subjects.

(G=2) Age Groups by AASM [2]

With sleep-specific age groups we refer to those suggested by the AASM scoring manual [2]. Indeed, it provides two sets of rules for visual sleep scoring, *i.e.*, babies/children and adults.
The age boundary between the two groups is not well defined; in particular, it is not clear which group the subjects in the age range between 13 and 18 (adolescents) belong to. Therefore we started considering two groups plus the adolescents' group: Babies/Children (CH; 0-12 years), Adolescents (A; 13-18 years), Adults (AD; $<$ 19 years).

In Supplementary Figure B.11 and in Table A.3, respectively, we show the age distribution of the *BSDB* dataset in the three age groups, and we report the number of recordings, the age range, the age mean and standard deviation ($\mu \pm \sigma$) and the male/female percentage (M/F) for each age group. We used U-sleep to evaluate if the PSGs of the Adolescents were closer to the Babies/Children's recordings or to

TABLE A.1: **Age groups by [105] overview with demographic statistics.**

Age groups	Recordings	Age in years (range)	Age in years ($\mu \pm \sigma$)	Sex % (F/M) *
B	151	0-3	1.6 ± 1.1	44/56
C	246	4-12	7.8 ± 2.6	43/57
A	177	13-17	15.3 ± 1.4	41/59
YA	2106	18-39	29.7 ± 6.3	42/58
MA	3655	40-59	50.4 ± 5.5	31/69
E	1546	60-69	64.0 ± 2.8	30/70
OE	988	70-91	75.1 ± 4.1	30/70

TABLE A.2: **Sleep parameters.**

	TST (min)	SPT (min)	WASO (%)	SL (min)	SE	n shift (n/h)	pN1 (%)	pN2 (%)	pN3 (%)	pREM (%)
all	340.2 ± 83.7	402.0 ± 71.4	15.7 ± 13.6	18.1 ± 25.1	84.3 ± 13.6	20.8 ± 7.8	20.2 ± 15.1	44.6 ± 14.2	17.9 ± 12.3	14.5 ± 7.6
B	452.0 ± 85.8	539.6 ± 88.5	15.8 ± 11.4	23.5 ± 30.4	84.2 ± 11.4	14.7 ± 5.4	13.0 ± 10.0	32.6 ± 14.0	29.4 ± 10.9	23.9 ± 9.3
C	430.2 ± 73.3	465.9 ± 75.9	7.5 ± 7.4	20.3 ± 28.3	92.5 ± 7.4	12.9 ± 4.1	8.0 ± 6.6	36.5 ± 11.6	36.5 ± 13.4	17.0 ± 6.9
A	377.8 ± 79.3	415.2 ± 85.0	9.0 ± 10.6	18.7 ± 30.1	91.0 ± 10.6	14.7 ± 5.2	8.9 ± 6.7	41.0 ± 11.1	32.1 ± 10.9	15.1 ± 6.3
YA	371.1 ± 93.9	414.9 ± 90.8	10.9 ± 11.4	16.1 ± 22.4	89.1 ± 11.4	18.1 ± 6.5	14.0 ± 10.9	44.4 ± 13.3	21.4 ± 10.9	15.8 ± 7.3
MA	336.3 ± 66.8	393.7 ± 55.4	14.7 ± 12.1	16.7 ± 22.8	85.3 ± 12.1	21.8 ± 7.4	20.1 ± 13.3	46.5 ± 13.8	16.1 ± 10.9	14.5 ± 7.3
E	311.4 ± 72.6	389.6 ± 56.8	20.3 ± 14.7	20.0 ± 26.5	79.7 ± 14.7	22.8 ± 8.1	25.3 ± 15.9	45.1 ± 14.6	14.5 ± 11.9	13.0 ± 7.5
OE	287.8 ± 73.4	385.6 ± 53.2	25.7 ± 15.7	22.6 ± 32.0	74.3 ± 15.7	23.4 ± 8.3	31.9 ± 19.3	41.5 ± 15.8	13.8 ± 12.2	11.6 ± 7.4

the Adults' recordings. We run two different experiments on *U-Sleep-v1*: in experiment_1 we merge the recordings from the Adolescents with the Babies/Children; in experiment_2 we merge the recordings from Adolescents with the Adults. In both the experiments (1, 2), we fine-tuned two different models (a, b), resulting in four independently trained models: (1a) fine-tuning on $G_{1a} = \{CH \cup A\}$; (1b) fine-tuning on $G_{1b} = \{AD\}$; (2a) fine-tuning on $G_{2a} = \{CH\}$; (2b) fine-tuning on $G_{2b} = \{A \cup AD\}$. For each experiment, we tested both the models (a) and (b) on the test set of the three groups $\{CH, A, AD\}$. In Table A.4 we report the macro F1-score (%F1), specifically the mean value and the standard deviation ($\mu \pm \sigma$), computed across the recordings. In bold we indicate the best performance achieved on each test set. We compared with a paired t-test the performance of the four models tested

TABLE A.3: **Age groups by AASM [2] overview with demographic statistics.** * The percentage sex % (F/M) has been computed on different percentage (from 99.4% up to 99.5%) of the total recordings for each age group, given the the lack of availability of the gender information.

Age groups	Recordings	Age in years (range)	Age in years ($\mu \pm \sigma$)	Sex % (F/M) *
CH	397	0-12	5.4 ± 3.7	43/57
A	177	13-17	15.3 ± 1.4	41/59
AD	8295	18-91	50.6 ± 15.6	34/66

TABLE A.4: ***U-Sleep-v1* F1-score on** $\{CH, A, AD\}$**.** Performance of *U-Sleep-v1* fine-tuned on $\{CH, A, AD\}$. We report the F1-score (%F1), specifically the mean value and the standard deviation ($\mu \pm \sigma$) computed across the recordings.
Best shown in bold.

Test Subsets	Experiment 1		Experiment 2	
	(1a) {CH A}	(1b) {AD}	(2a) {CH}	(2b) {A AD}
CH	**75.3 ± 8.0**	68.2 ± 12.4	**75.4 ± 7.9**	71.8 ± 10.2
A	76.5 ± 19.1	**82.9 ± 13.7**	78.4 ± 18.1	**82.8 ± 13.6**
AD	72.1 ± 13.0	**77.7 ± 10.9**	72.2 ± 13.3	**77.5 ± 10.8**

on the Adolescents. The model (2b), fine-tuned on $\{A \cup AD\}$, performs significantly better ($p - value < 0.05$) on the Adolescents than the model (1a), fine-tuned on $\{CH \cup A\}$, suggesting that Adolescents tend to be similar similar to Adults. This is confirmed by the fact that also the model (1b), fine-tuned on $\{AD\}$, performs better ($p - value < 0.05$) on the Adolescents than the model (1a), even without Adolescents' recordings in the training set. The models (1b) and (2b), fine-tuned on Adults without or with Adolescents, reach the same performance (paired t-test $p - value > 0.05$), hence confirming again the statement above. We might conclude that the Adolescents belong to the Adults' group, so as to run all the age conditioning analysis on the following two sleep-related age groups: $G_1 = \{B \cup C\}$ and $G_2 = \{A \cup YA \cup MA \cup E \cup OE\}$.
For both $\{CH\}$ and $\{AD\}$ we obtain the same performance (paired t-test $p - value > 0.05$) with the two models fine-tuned on the group itself, with or without Adolescents. However, we reached significantly lower performance (paired t-test $p - value < 0.01$) with the other two models fine-tuned on the complementary group. This latter statement strengthens the basic assumption that babies/children and adults are two different groups for the DL scoring algorithm.

In Supplementary Figure B.22 we report the confusion matrix for each of the five models (0, 1a, 1b, 2a, 2b) and each of the three test sets $\{CH, A, AD\}$. With model (0) we refer to the model fine-tuned on the whole training set, regardless of the subjects' age.

A.2 U-Sleep: Consciousness detection in AS/DS children

In Chapter 6 we demonstrate the resilience of U-Sleep, a state-of-the-art sleep scoring architecture, against the AASM guidelines. U-Sleep has been evaluated on tens of thousands of PSGs from different large-scale-heterogeneous data cohorts. In order to rationalize the analyses that follow, we want to highlight the main peculiarities of U-Sleep and the reasons why we choose this architecture: it processes inputs of arbitrary length, from any arbitrary EEG electrode positions, from any hardware and software filtering; it predicts the sleep stages for the whole PSG in a single forward pass; it outputs sleep stage labels at any temporal frequency, up to the signal sampling rate, *i.e.*, it can label sleep stages at shorter intervals than the standard 30-seconds, up to one sleep stage per each sampled time point.

In this supplementary section, the ultimate goal is to demonstrate that U-Sleep, if properly pre-trained, it is actually also able to recognize consciousness (awake) and unconsciousness (non-rapid eye movement - NREM) states under conditions of abnormal cortical dynamics (*i.e.*, abnormal sleep physiology), by only looking at the EEG activity. Hence, we first pre-train *U-Sleep-v1* on CHAT and NCH datasets (*i.e.*, whole night recordings from healthy children and children with different sleep disorders) to solve the binary classification task: awake vs NREM states. Then we evaluate the pre-trained architecture on children with Angelman Syndrome (AS) and Dup15q (duplication of the chromosome 15q11.2-13.1) Syndrome (DS), two genetic disorders characterized by abnormal sleep physiology and abnormal EEG rhythms. Children with Angelman Syndrome (AS) show an unusual delta EEG activity. It may resemble slow wave sleep activity during wakefulness, but they are clearly conscious while awake. Children with Dup15q Syndrome (DS) show an unusual beta EEG activity. It may resemble wakefulness during sleep. Thus, conscious (awake) and unconscious (NREM) states clearly become difficult to score by only looking at the EEG activity.

We exploit 4522 children whole-night recordings from the two publicly available clinical studies CHAT and NCH.

CHAT. The Childhood Adenotonsillectomy Trial database consists of 1638 recordings (452 baseline, 407 follow-up and 779 control) from 1232 children post-adenotonsillectomy-surgery aged 5-10 years. The recordings are collected in six different sleep centers in Massachusetts, Missouri, New York, Ohio and Pennsylvania [36], [40], [41]. EEG signals (F4-M1, F3-M2, C4-M1, C3-M2, O2-M1, O1-M2, T4-M1, T3-M2) have been considered in our experiments. The signals are recorded at 200Hz (or higher in other sleep centers), and different hardware filtering given the different acquisition systems. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to `https://doi.org/10.25822/d68d-8g03` and `https://clinicaltrials.gov/ct2/show/NCT00560859`.

NCH. The Nationwide Children's Hospital Sleep DataBank [36], [123]. The database contains pediatric sleep studies of 3673 patients conducted at NCH in Columbus, Ohio, USA. We consider a total of 2884 sleep recordings from 2674 children aged 0-12 years, as all the other recordings were excluded due to chronological age outside the children class-range (see section A.1), to missing data, inability to align PSG recordings with the hypnogram events, and other technical problems. EEG signals (F4-M1, F3-M2, C4-M1, C3-M2, O2-M1, O1-M2) have been considered in our experiments. The signals are recorded at one of three sampling rates: 250Hz,

400Hz, or 512Hz, and different hardware filtering given the different acquisition systems. The recordings are manually scored by sleep experts according to the AASM rules. For more information we refer to https://sleepdata.org/datasets/nchsdb.

We pre-train *U-Sleep-v1* on both the CHAT and the NCH datasets. We split each open access dataset per-subjects in training (80%) and validation (20%). After pre-trainig U-Sleep, we evaluate the pre-trained architecture on AS and DS datasets.

AS. Children with Angelman Syndrome are recruited through an NIH funded AS Natural History Study [NCT00296764]. The EEG signals are recorded at two Boston Children's Hospital and Rady Children's Hospital San Diego as part of the study during wakefulness and sleep in a clinical setting. EEG signals (F4, F3, C4, C3, T3, T4, O2, O1) have been considered in our experiments. The signals are recorded at one of three sampling rates: 250Hz, 256Hz, or 512Hz. Periods of drowsiness, sleep and awake are delineated by the EEG technician during recordings using data annotations, and EEG annotations are reviewed by a certified neurologist. Note that because the AS EEG phenotype generally resembles slow wave sleep, sleep scoring of AS EEG into specific NREM stages (i.e., N1, N2, or N3) is often unreliable, and wake EEG can potentially be mislabeled as sleep EEG without annotations provided by an EEG technician. We used 25 EEG recordings from 21 children, aged 1-10 years. For more information we refer to [124].

DS. Children with Dup15q Syndrome are recruited through the University of California, Los Angeles, (UCLA), at the Department of Psychology. The EEG signals are recorded at the UCLA Ronald Reagan Medical Center. EEG signals (F4, F3, C4, C3, T3, T4, O2, O1) have been considered in our experiments. The signals are recorded at a sampling rate of 200 Hz. Sections of N2 sleep were delineated by sleep splines, which were automatically detected using the Python-based toolbox YASA (Yet Another Spindle Algorithm) [109]. 30-minute sections of wake recordings during mid-to-late afternoon are extracted. Wakefulness is inferred by the presence of ocular (*e.g.*, blink or saccades) and EMG artifacts in data. A certified neurologist reviewed all extracted EEG sections to confirm that they were scored correctly as wake or NREM sleep. We used 22 EEG recordings from 11 children, aged 0-11 years.

The data pre-processing and EEG data selection/sampling across all the datasets is the same as in the previous Chapter 6. The only changes here are that we are using a single channel EEG, and we are considering only the two labels awake versus NREM (*i.e.*, we have merged the N1, N2 and N3 stages into a single stage NREM, and we have excluded the MOVEMENT, REM and UNKNOWN classes). The binary loss function for stages as MOVEMENT, REM and UNKNOWN is masked during the training procedure.

The AS and DS data has been further preprocessed as done in [125]. We first lowpass filter EEG signals at 45 Hz using a finite impulse response filter with the filter order selected as twice the sampling rate of the signal. Next, we highpass filter EEG signals at 0.4 Hz using a 5th order Butterworth filter; the stopband attenuation and roll-off of this filter were optimal for attenuating drift artifacts while minimally attenuating slow oscillations $\geq$ 0.5 Hz (0.44 dB attenuation at 0.5 Hz). Following filtering, each EEG channel is re-referenced to average. Next, we manually exclude EEG sections with gross physiological and technical artifacts. Periods of flickering light stimulation intended to trigger epileptiform activity in participants with AS

TABLE A.5: ***U-Sleep-v1* consciousness detection performance on AS/DS children.** Performance of *U-Sleep-v1*, pre-trained on the open access CHAT and NCH datasets, and evaluated on AS and DS datasets. We report the weighted F1-score (%F1) and the per-class F1-score. The metrics refer to a high frequency evaluation of the sleep states - prediction for each one-second window.

Datasets	F1	F1/awake	F1/NREM
AS	75.0	55.8	83.0
DS	94.7	94.7	95.7

are also excluded. Next, we mark noisy channels to be excluded from independent component analysis (ICA) and later interpolated following data cleaning. ICA was then used (FastICA algorithm) to remove stereotyped artifacts such as EMG and eye movements [126]. Finally, we spatially interpolate noisy channels and repeated average referencing. EEG datasets are rejected if they did not yield at least 15 valid frequency transform windows for 0.5 Hz, i.e., the lowest frequency analyzed in sleep scoring.

We evaluate U-Sleep as stated in [57]. The model scores the full PSG, without considering the predicted class on a segment with a label different from the five sleep stages (*e.g.*, segment labelled as 'UNKNOWN' or as 'MOVEMENT'). The final prediction is the results of all the possible combinations of the available EEG channels for each recordings. Hence, we use the majority vote, *i.e.*, the ensemble of predictions given by the multiple combination of channels in input. The AS and DS recordings has been segmented in windows of one-second. Therefore, we evaluate U-Sleep on both AS and DS recordings exploiting the high-frequency prediction property of the architecture, *i.e.*, it outputs one sleep stage per each one-second window.

Despite the different data domain (*e.g.*, different recording hardware) and the likely impact of the abnormal EEG activities, we achieved remarkable performance in terms of weighted F1-score on both the unseen datasets AS (F1-score 75.0%) and DS (F1 score 94.7%). As expected, on the AS dataset the slow delta waves during wakefulness are fooling the algorithm, *i.e.*, the model is mainly forecasting NREM state even when the subjects are awake. On the other hand, on the DS dataset the algorithm is robust against the fast beta waves during sleep, *i.e.*, the model is still able to recognize the two different states. We can conclude that a DL based algorithm, specifically U-Sleep, pre-trained on a huge and heterogeneous dataset with children aged 0-12 years, is able to generalize on a never seen dataset of children with genetic disorders characterized by abnormal sleep physiology.

Appendix B

Supplementary Tables and Figures

TABLE B.1: **Overview of the available deep learning based scoring architectures.** Summary of systems for sleep scoring using deep learning classification techniques directly applied to raw data. We report: the dataset from which the data were extracted; the type and number of subjects considered in the analysis; the type and number of channels taken into account; the type of deep learning classification algorithms and the best performance achieved. ANN: artificial neural network, CNN: convolutional neural network, EEG: electroencephalogram, EMG: electromyogram, EOG: electrooculogram (LOC/EOG1/E1, ROC/EOG2/E2: left, right EOG, respectively), GRU: gated recurrent unit, LSTM: long short-term memory, MLP: multilayer perceptron, MSLT: multiple sleep latency test, PD: Parkinson's disease, PSG: polysomnography, RCNN: recurrent-convolutional neural network, RNN: recurrent neural network, VGG: visual geometry group. *Acc.*: Accuracy; *Sens.*: Sensitivity; *Agr.*: computer scoring / visual scoring agreement.

Datasets	Dataset & Subjects	Channels	Classifier	Performance
Tsinalis et al. 2016 [58]	39 recordings (healthy) [1]	EEG single-channel (Fpz-Cz)	CNNs + 2D stack of frequency-specific activity in time (end-to-end ANN)	Validation set *Overall Acc.* 71-76% *Per-stage Acc.* 80-84%
Supratak et al. 2017 [29]	62 recordings (healthy) SS3 [2] 39 recordings (healthy) [1]	EEG single-channel (F4-EOG1 or Fpz-Cz or Pz-Oz)	Low-frequency information and high frequency information using CNNs + RNN (two bi-LSTM layers)	Validation set [2] *Acc.* 86.2% *Kappa* 0.80 Validation set [1] (SEDF-SC-13) *Acc.* 82.0% *Kappa* 0.76
Vilamala et al. 2017 [59]	39 recordings (20 healthy) [1]	EEG single-channel (Fpz-Cz)	Time-frequency image + CNN (VGGNet as VGG-FE feature extractor and as VGG-FT fine-tuned network)	Test set VGG-FE *Acc.* 84-88% VGG-FT *Acc.* 84-88%

Continued on next page

Datasets	Dataset & Subjects	Channels	Classifier	Performance
Biswal et al. 2018 [35]	10000 recordings [3] 5804 recordings [4]	EEG multi-channel (F3, F4, C3, C4, O1, O2) EEG multi-channel (C3, C4)	Spectrograms + RCNN (CNN + RNN)	Train on data[3] Test set[3] *Acc.* 87.5% *Kappa* 0.80 Test set[4] *Acc.* 77.7% *Kappa* 0.73
Chambon et al. 2018 [53]	61 recordings (healthy) SS3 [2]	EEG multi-channel (F3, F4, C3, C4, O1, O2) EOG1, EOG2 three chin EMG	Multivariate network architecture: linear spatial filtering + CNN	Test set *Sens.* 52%
Cui et al. 2018 [61]	116 recordings (healthy, sick, under treatment) [5]	EEG multi-channel (F3, F4, C3, C4, O1, O2) LOC, ROC X1, X2 and X3 EMG	CNN + fine-grained segment in multiscale entropy	Test set *Acc.* 92.2%
Malafeev et al. 2018 [54]	54 recordings (healthy) [6] 43 recordings (22 PSG and 21 MSLT narcolepsy and hypersomnia) [7]	EEG single-channel (Pz-Oz) one EMG two EOG	CNN (11 layers) + two bi-LSTM layers; Residual CNN (19 layers) + two bi-LSTM layers	Test set *Overall Kappa* 0.8 (except N1 with Kappa <0.5) *see paper for details*
Olesen et al. 2018 [62]	2310 recordings (healthy and patients) [8]	EEG multi-channel (central and occipital) EOG1, EOG2 chin EMG	Deep residual network model - 50 convolutional layers	Test set *Acc.* 84.1% *Kappa* 0.75

Continued on next page

Datasets	Dataset & Subjects	Channels	Classifier	Performance
Patanaik et al. 2018 [52]	1046 recordings DS1 (healthy adolescents) [9] 284 recordings DS2 (healthy young adults) [10] 210 recordings DS3 (sleep disorders) [11] 77 recordings DS4 (PD adults patients) [12]	EEG multi-channel (C3, C4) EOG (E1,E2)	Spectral Image + deep CNN + MLP stage classifier	Train on data [9, 10] Test set [9, 10] *Acc.* 89.8% *Kappa* 0.86 Validation set [11] *Acc.* 81.4% *Kappa* 0.74 Validation set [12] *Acc.* 72.1% *Kappa* 0.60
Sors et al. 2018 [63]	5793 recordings (patients) [4]	EEG single-channel (C4-A1)	14 layers CNN	Test set *Acc.* 87% *Kappa* 0.81
Stephansen et al. 2018 [67]	3000 recordings (healthy and patients) from over 10 databases	EEG multi-channel (C3 or C4 and O1 or O2) LOC, ROC chin EMG	CNN + RNN	Test set on IS-RC *Acc.* 87% *see paper for details*
Zhang and Wu 2018 [60]	25 recordings (sleep-disordered breathing) [13] 16 recordings [14]	EEG single-channel	Complex-valued unsupervised CNN	Train on data [13] Validation set [13] *Acc.* 87% *Kappa* 0.81 Test set [14] *Acc.* 87.2%
Perslev et al. 2019 [56]	153 recordings (healthy) [1] 994 recordings (sleep disorders) [15] 255 recordings (sleep disorders) [16] 99 recordings (sleep disorders) [5] 25 recordings (sleep-disordered breathing) [13]	EEG single-channel (Fpz-Cz or C3-A2)	CNNs U-Net-based architecture	Test set [1] (SEDF-SC-13) *MF1.* 0.79 Test set [1] (SEDF-SC-18)) *MF1.* 0.76 Test set [15] *MF1.* 0.77 Test set [16] *MF1.* 0.79 Test set [5] *MF1.* 0.77 Test set [13] *MF1.* 0.73

Continued on next page

Datasets	Dataset & Subjects	Channels	Classifier	Performance
Phan et al. 2019 [66]	200 recordings (healthy) [2]	EEG single-channel (C4-A1) EOG1, EOG2 two chin EMG	Time-frequency images + end-to-end hierarchical RNN for sequence-to-sequence sleep staging	Test set *Acc.* 87.1% *Kappa* 0.81
Mousavi et al. 2019 [69]	153 recordings (healthy) [1]	EEG single-channel (Fpz-Cz or Pz-Oz)	Low-frequency information and high frequency information using CNNs + Encoder-Decoder RNN (bi-LSTM layers)	Validation set [1] (SEDF-SC-13) *Acc.* 84.3 *Kappa* 0.79 Validation set [1] (SEDF-SC-18)) *Acc.* 80.0 *Kappa* 0.73
Yildirim et al. 2019 [64]	Eight recordings (healthy, mild difficulty in falling asleep) [17] 61 recordings (healthy and mild difficulty in falling asleep) [1]	EEG single-channel (Fpz-Cz) single horizontal EOG channel	CNN	Test set [17] *Acc.* 91.22% Test set [1] *Acc.* 90.98%
Fiorillo et al. 2020 [7]	39 recordings (healthy) [1]	EEG single-channel (Fpz-Cz)	Low-frequency information and high frequency information using CNNs + RNN (one bi-LSTM layer)	Validation set [1] (SEDF-SC-13) *Acc.* 85.2 *Kappa* 0.80
Guillot et al. 2020 [25]	25 recordings (healthy) [18] 55 recordings (sleep apnea) [19]	EEG multi-channel (C3-M2, C4-M1, F4-M1, F3-F4, F3-M2, F4-O2, F3-O1, FP1-F3, FP1-M2, FP1-O1, FP2-F4, FP2-M1, FP2-O2, O1-M2, O2-M1)	Time-frequency image + bi-GRU with Attention Layer + Positional Embedding + bi-GRU with skip-connection sequence encoder	Test set [18] *Acc.* 89.9 *Kappa* 0.85 Test set [19] *Acc.* 88.7 *Kappa* 0.82

Continued on next page

Datasets	Dataset & Subjects	Channels	Classifier	Performance
Seo et al. 2020 [70]	39 recordings (healthy) 1 62 recordings (healthy) SS3 2 5791 recordings (patients) 4	EEG single-channel (Fpz-Cz or F4-EOG1 or C4-A1)	CNNs + RNN (two bi-LSTM layers)	Test set 1 (SEDF-SC-13) *Acc.* 83.6 *Kappa* 0.77 Test set 2 *Acc.* 86.2 *Kappa* 0.79 Test set 4 *Acc.* 86.3 *Kappa* 0.81
Supratak et al. 2020 [72]	200 recordings (healthy) 2 153 recordings (healthy) 1	EEG single-channel (F4-EOG1 or Fpz-Cz)	Low-frequency information and high frequency information using CNNs + RNN (one bi-LSTM layer)	Test set 2 *Kappa* SS1 0.76 *Kappa* SS2 0.75 *Kappa* SS3 0.82 *Kappa* SS4 0.77 *Kappa* SS5 0.81 Test set 1 (SEDF-SC-13) *Acc.* 85.4 *Kappa* 0.80 Test set 1 (SEDF-SC-18)) *Acc.* 83.1 *Kappa* 0.77
Guillot et al. 2021 [107]	5788 recordings from 7 clinical studies	EEG multi-channel EOG1, EOG2 EMG	Time-frequency image + bi-GRU with Attention Layer + Positional Embedding + bi-GRU with skip-connection sequence encoder	Test set MF1 score on average 0.78 *see paper for details*
Olesen et al. 2021 [108]	15684 recordings from five clinical studies	EEG multi-channel (C3-M2, C4-M1) EOG1, EOG2 chin EMG	Deep residual network model - ResNet-50-based architecture	Test set overall accuracy on average 0.82 *see paper for details*

Continued on next page

Datasets	Dataset & Subjects	Channels	Classifier	Performance
Phan et al. 2021 [73]	153 recordings (healthy) [1] 200 recordings (healthy) [2] 994 recordings (sleep disorders) [16] 5791 recordings (patients) [4]	EEG single-channel (Fpz-Cz or C4-A1 or C3-A2) EOG (ROC-LOC or E1-M2) EMG (chin1-chin2)	Two parallel neural networks: fully CNNs and attention-based RNNs + bi-directional RNNs (LSTM and GRU cells)	Test set [1] (SEDF-SC-13) *Acc.* 86.4 *Kappa* 0.81 Test set [1] (SEDF-SC-18)) *Acc.* 83.9 *Kappa* 0.77 Test set [2] *Acc.* 87.6 *Kappa* 0.82 Test set [16] *Acc.* 81.4 *Kappa* 0.75 Test set [4] *Acc.* 89.1 *Kappa* 0.85
Perslev et al. 2021 [57]	19924 recordings from 16 clinical studies	EEG single-channel and EOG single-channel (all possible combination of derivations)	CNNs U-Net-based architecture	Test set F1-dice score on average 0.79 *see paper for details*

Databases: Sleep-EDF ✓[Expanded] SEDF-SC-13 and SEDF-SC-18 [1]; Montreal archive of sleep studies MASS (✓) (SS1-SS5) [2]; Massachusetts General Hospital (MGH) Sleep Laboratory [3]; Sleep Heart Health Study (SHHS) (✓) [4]; ISRUC-sleep ✓[5]; University of Zurich [6]; Psychiatry and Neurology in Warsaw [7]; Wisconsin Sleep Cohort [8]; CNL lab, Singapore [9]; CSL lab, Singapore [10]; SDU, Singapore GH [11]; UC San Diego sleep lab [12]; The St. Vincent's University Hospital; University College Dublin Sleep Apnea Database (SVUH-UCD) ✓[13]; MIT-BIH database ✓[14]; PhysioNetCinC Challenge by the Massachusetts General Hospital's Computational Clinical Neurophysiology Laboratory and the Clinical Data Animation Laboratory (PHYS) ✓[15]; The Danish Centre for Sleep Medicine (DCSM) ✓[16]; Sleep-EDF ✓[17]; Dreem Open Dataset - Healthy (DOD-H) ✓[18]; Dreem Open Dataset - Obstructive Sleep Apnea (DOD-O) ✓[19]. Datasets directly available online are identified by ✓, whilst datasets that require approval from a Data Access Committee marked by (✓).

TABLE B.2: **DSN-L models performance after query procedure w/o *MC*.** Overall performance of the models obtained from 20-fold and 10-fold cross-validation without *MC* on both SEDF-SC-13 ±30mins and SEDF-SC-18 ±30mins datasets. The metrics refer to the epochs kept after the standard (*i.e.*, *w/o MC* the predicted probability values $max(\hat{\boldsymbol{p}}_i)$ used to select the uncertain instances) query selection procedure (q% threshold value fixed to 5%). We report the overall accuracy (%Acc.), macro F1-score (%MF1), Cohen's Kappa (k), weighted-averaging F1-score (%F1), averaged $confidence$ value and percentage of misclassified epochs among the rejected with query (%miscl.). The best performance metrics for each dataset are indicated in bold.

			Overall Metrics				
	Datasets	Models	Acc.	MF1	k	F1	%miscl.
w/o MC	SEDF-SC-13 ±30mins	*base*	84.3%	78.2%	0.78	84.4%	54.1%
		base+LS_U	**85.0%**	**79.1%**	**0.79**	**85.1%**	**57.4%**
		base+LS_S	84.8%	78.0%	0.79	84.8%	56.6%
	SEDF-SC-18 ±30mins	*base*	**81.4%**	**76.0%**	**0.75**	**81.9%**	57.8%
		base+LS_U	81.2%	75.8%	0.74	81.6%	56.2%
		base+LS_S	81.2%	75.9%	0.74	81.7%	**58.6%**

TABLE B.3: **DSN-L models performance after query procedure w/ *MC*.** Overall performance of the models obtained from 20-fold and 10-fold cross-validation without *MC* on both SEDF-SC-13 ±30mins and SEDF-SC-18 ±30mins datasets. The metrics refer to the epochs kept after both the ${\sigma^2}_{\mu\max}$ and $\mu_{\max}$ query selection procedures (q% threshold value fixed to 5%). We report the overall accuracy (%Acc.), macro F1-score (%MF1), Cohen's Kappa (k), weighted-averaging F1-score (%F1), averaged $confidence$ value and percentage of misclassified epochs among the rejected with query (%miscl.). The best performance metrics for each dataset are indicated in bold.

				Overall Metrics				
	Datasets		Models	Acc.	MF1	k	F1	%miscl.
w/ MC	SEDF-SC-13 ±30mins	${\sigma^2}_{\mu\max}$	*base*	84.7%	**78.3%**	0.79	84.6%	**49.7%**
			base+LS_U	**85.7%**	77.9%	**0.80**	**85.2%**	47.0%
			base+LS_S	84.9%	78.2%	0.79	84.8%	44.9%
		$\mu_{\max}$	*base*	85.2%	78.9%	0.80	85.2%	**59.2%**
			base+LS_U	**86.1%**	**79.6%**	**0.81**	**86.0%**	55.2%
			base+LS_S	85.5%	78.6%	0.80	85.4%	57.3%
	SEDF-SC-18 ±30mins	${\sigma^2}_{\mu\max}$	*base*	**81.7%**	**76.7%**	**0.75**	**82.1%**	**45.1%**
			base+LS_U	81.7%	75.9%	0.75	81.8%	44.8%
			base+LS_S	81.6%	76.3%	0.75	81.9%	41.8%
		$\mu_{\max}$	*base*	**82.4%**	**76.9%**	**0.76**	**82.7%**	**58.2%**
			base+LS_U	82.3%	76.7%	0.76	82.5%	57.2%
			base+LS_S	82.4%	76.8%	0.76	82.7%	57.8%

TABLE B.4: **DSN-L models performance +LS_{SC} after query procedure w/o *MC*.** Overall performance of the DSN-L models on IS-RC, DOD-H and DOD-O datasets. The metrics refer to the epochs kept after the standard (*i.e.*, *w/o MC* the predicted probability values $max(\hat{\boldsymbol{p}}_i)$ used to select the uncertain instances) query selection procedure (q% threshold value fixed to 5%). We report the overall accuracy (%Acc.), macro F1-score (%MF1), Cohen's Kappa (k), weighted-averaging F1-score (%F1) and percentage of misclassified epochs among the rejected (%miscl.). The best performance metrics for each dataset are indicated in bold.

			Overall Metrics				
Datasets	Models	α	Acc.	MF1	k	F1	%miscl.
IS-RC	*base*	-	79.2	69.6	0.69	79.7	56.7
	base+LS_U	0.4	79.4	69.9	0.70	80.0	58.6
	base+LS_{SC}	0.6	**81.6**	**72.0**	**0.72**	**82.0**	**60.2**
DOD-H	*base*	-	78.6	71.6	0.70	78.8	54.4
	base+LS_U	0.2	76.9	70.2	0.68	76.7	55.1
	base+LS_{SC}	0.8	**82.1**	**74.3**	**0.74**	**82.3**	**56.1**
DOD-O	*base*	-	71.2	51.7	0.58	71.6	**60.4**
	base+LS_U	0.1	76.6	**57.8**	0.65	77.5	57.6
	base+LS_{SC}	1	**77.7**	57.4	**0.66**	**77.9**	59.5

TABLE B.5: **Atypical and/or randomly ordered channel derivations.** U-Sleep channel extraction for each open database: (*U-Sleep-v0*) atypical and/or randomly ordered channel derivations are extracted from the available channels; (*U-Sleep-v1*) correctly ordered channel derivations are extracted from the available channels, i.e., expected clinical derivations meant to be extracted in [57].

Datasets	Channel type	*U-Sleep-v0*	*U-Sleep-v1*
ABC	EEG	F3-F4	F3-M2
		O1-C3	F4-M1
		C4-F4	C3-M2
		E2-C3	C4-M1
		O2-F4	O1-M2
		M1-C3	O2-M1
	EOG	M2-F4	E1-M2
		E1-C3	E2-M1
CCSHS	EEG	LOC-C4	C3-A2
		M2-ROC	C4-A1
	EOG	M1-C4	LOC-A2
		C3-ROC	ROC-A1
CFS	EEG	A2-A1	C3-A2
		C3-C4	C4-A1
	EOG	ROC-A1	LOC-A2
		LOC-C4	ROC-A1
CHAT*	EEG	M2-E1	F3-M2
		F4-T4	F4-M1
		C3-E1	C3-M2
		E2-T4	C4-M1
		C4-E1	T3-M2
		T3-T4	T4-M1
		M1-E1	O1-M2
		O2-T4	O2-M1
	EOG	O1-E1	E1-M2
		F3-T4	E2-M1
DCSM	EEG	F3-M2	F3-M2
		F4-M1	F4-M1
		C3-M2	C3-M2
		C4-M1	C4-M1
		O1-M2	O1-M2
		O2-M1	O2-M1
	EOG	E1-M2	E1-M2
		E2-M2	E2-M2
HPAP **	EEG	-	F3-M2
		-	F4-M1
		-	C3-M2

Continued on next page

Datasets	Channel type	*U-Sleep-v0*	*U-Sleep-v1*
		-	C4-M1
		-	O1-M2
		-	O2-M1
	EOG	-	E1-M2
		-	E2-M2
MESA	EEG	E2-Fpz	Fpz-Cz
		C4-M1	Cz-Oz
		E1-Fpz	C4-M1
	EOG	Fz-Cz	E1-Fpz
		Cz-Oz	E2-Fpz
MROS	EEG	E1-C4	C3-M2
		M1-C3	C4-M1
	EOG	M2-C4	E1-M2
		E2-C3	E2-M1
PHYS	EEG	F3-M2	F3-M2
		F4-M1	F4-M1
		C3-M2	C3-M2
		C4-M1	C4-M1
		O1-M2	O1-M2
		O2-M1	O2-M1
	EOG	E1-M2	E1-M2
SEDF-SC	EEG	Pz-Oz	Fpz-Cz
		Fpz-Cz	Pz-Oz
	EOG	EOG	EOG
SEDF-ST	EEG	Pz-Oz	Fpz-Cz
		Fpz-Cz	Pz-Oz
	EOG	EOG	EOG
SHHS	EEG	C4-A1	C4-A1
		C3-A2	C3-A2
	EOG	EOGL-PG1	EOGL-PG1
		EOGR-PG1	EOGR-PG1
SOF	EEG	LOC-A2	C3-A2
		A1-C4	C4-A1
	EOG	C3-A2	LOC-A2
		ROC-C4	ROC-A1

* The CHAT dataset has recordings where we may find a different order of EEG and EOG sensors for different edf files. Consequently, in the U-Sleep version where they were erroneously extracting atypical and/or randomly ordered channel derivations (U-Sleep-v0), we can generate multiple combinations of incorrect derivations. In Table we report the most frequent incorrect EEG and EOG derivations.

** The HPAP dataset has recordings where we can find a different order of EEG and EOG sensors for each edf file, resulting in different combinations of incorrect derivations for each recording. For that reason, we preferred not to report the incorrect and completely random combinations of derivations between the different recordings.

TABLE B.6: **Experiments *(i)*: *U-Sleep-v1* weighted-F1-score.**
(i) Performance of *U-Sleep-v0* and *U-Sleep-v1*, pre-trained on the open access (OA) datasets, and evaluated on the test set of the *BSDB* dataset, and on the whole $BSDB_{(100\%)}$ dataset, *i.e.*, both direct transfer (DT) on *BSDB*. We report the weighted F1-score (%wF1), specifically the mean value and the standard deviation $(\mu \pm \sigma)$ computed across the recordings.

	Training on OA	
Datasets	*U-Sleep-v0*	*U-Sleep-v1*
BSDB	77.8 ± 10.9	77.9 ± 10.8
$BSDB_{(100\%)}$	78.2 ± 11.1	78.3 ± 11.2

TABLE B.7: **Experiments *(ii)*: *U-Sleep-v1* weighted-F1-score.**
(ii) Performance of *U-Sleep-v1*, pre-trained on the open access (OA) datasets, and evaluated on all the test set of the open access datasets and on the test set of the *BSDB* dataset. We also report the performance of *U-Sleep-v1* trained from scratch (S) or fine-tuned (FT) on the *BSDB* dataset, and evaluated on all the test set of all the available datasets. We report the weighted F1-score (%wF1), specifically the mean value and the standard deviation $(\mu \pm \sigma)$ computed across the recordings.

	Training on OA	Training on *BSDB*	
Datasets	*U-Sleep-v1*	*U-Sleep-v1* (S)	*U-Sleep-v1* (FT)
ABC	81.3 ± 8.5	78.9 ± 10.1	76.5 ± 10.1
CCSHS	90.3 ± 4.6	85.3 ± 6.3	85.4 ± 5.9
CFS	87.8 ± 6.6	82.2 ± 7.9	82.8 ± 7.2
CHAT	86.4 ± 4.8	79.3 ± 6.7	76.5 ± 7.5
DCSM	90.5 ± 4.4	81 ± 8.9	79.1 ± 9.7
HPAP	80.6 ± 7.5	75.8 ± 9.6	74.5 ± 11.7
MESA	84.2 ± 7.2	79.1 ± 12.9	79.6 ± 9.3
MROS	85.3 ± 7.0	75.5 ± 10.5	77.4 ± 9.8
PHYS	80.5 ± 8.6	79.2 ± 8.9	79.2 ± 9.0
SEDF-SC	86.7 ± 5.5	85.1 ± 5.6	86.6 ± 5.3
SEDF-ST	83.5 ± 4.5	73.6 ± 8.3	74.9 ± 6.4
SHHS	86.6 ± 6.3	81.6 ± 7.1	83.5 ± 6.5
SOF	85.6 ± 6.5	75.6 ± 10.9	78.4 ± 9.9
avg OA	85.7 ± 7.1	79.7 ± 9.4	80.0 ± 9.0
BSDB	77.9 ± 10.8	82.2 ± 9.4	82.0 ± 9.4

TABLE B.8: **Experiments *(iii)*: *U-Sleep-v1* weighted-F1-score.** *(iii)* Performance of *U-Sleep-v1* on a single model fine-tuned on all the training set of the seven *BSDB* groups (FT); on seven/two models fine-tuned on the independent training set of each group (FT-I) with $G = 7$ and $G = 2$ respectively; and on a single model fine-tuned on all the training set of the seven/two *BSDB* groups conditioned (FT-SaBN) by $G = 7$ and by $G = 2$ groups respectively. All the fine-tuned models are evaluated on the associated test set of each group. We report the weighted F1-score (%wF1), specifically the mean value and the standard deviation ($\mu \pm \sigma$) computed across the recordings. B: Babies (0-3 years); C: Children (4-12 years); A: Adolescents (13-18 years); YA: Young Adults (19-39 years); MA: Middle Aged Adults (40-59 years); E: Elderly (60-69 years); OE: Old Elderly (> 70 years). When $G = 2$ we have the following two groups $G_1 = \{B \cup C\}$, $G_2 = \{A \cup YA \cup MA \cup E \cup OE\}$.

Subsets	FT	FT-I		FT-SaBN	
	(G=1)	(G = 7)	(G = 2)	(G = 7)	(G = 2)
B	79.2 ± 7.4	78.7 ± 7.2 ^{G_1}	77.6 ± 8.5 ^{G_1}	79.5 ± 6.7	77.2 ± 7.9
C	80.8 ± 9.5	80.7 ± 8.9 ^{G_2}	81.1 ± 7.7 ^{G_1}	81.5 ± 8.0	81.4 ± 7.9
A	87.3 ± 10.8	86.0 ± 13.4 ^{G_3}	87.0 ± 10.7 ^{G_1}	87.0 ± 10.8	86.2 ± 11.3
YA	85.9 ± 9.3	85.6 ± 9.4 ^{G_4}	85.4 ± 9.5 ^{G_2}	85.5 ± 9.4	84.6 ± 9.8
MA	84.1 ± 6.2	83.5 ± 6.6 ^{G_5}	83.4 ± 6.4 ^{G_2}	83.5 ± 6.6	82.9 ± 6.9
E	80.5 ± 8.3	79.4 ± 9.0 ^{G_6}	79.6 ± 8.9 ^{G_2}	80.0 ± 8.2	79.0 ± 9.4
OE	79.8 ± 9.6	78.9 ± 9.4 ^{G_7}	79.1 ± 9.7 ^{G_2}	79.4 ± 9.5	78.8 ± 9.4
avg	82.5 ± 9.0	81.8 ± 9.4	81.9 ± 9.2	82.2 ± 8.9	81.4 ± 9.34

TABLE B.9: ***U-Sleep-v1* weighted-F1-score and uncertainty estimate.** Performance of *U-Sleep-v1*, pre-trained on the open access (OA) datasets, and evaluated on all the test set of the open access datasets (*avg* OA) with and without (*i.e.*, *U-Sleep-v1* pre-trained in *(ii)*) label smoothing. We report the weighted F1-score (%F1), referred to the epochs kept after the $\mu_{\max}$ query selection procedure (q% threshold value fixed to 5%), and we report percentage of misclassified epochs among the rejected with query (%miscl.). Specifically, we report the mean value and the standard deviation ($\mu \pm \sigma$) computed across the recordings.

Dataset		*w/o* label smoothing	*w/* label smoothing
avg OA	%F1	87.4 ± 7.3	82.5 ± 9.8
	%miscl.	51.1 ± 9.5	53.7 ± 10.6

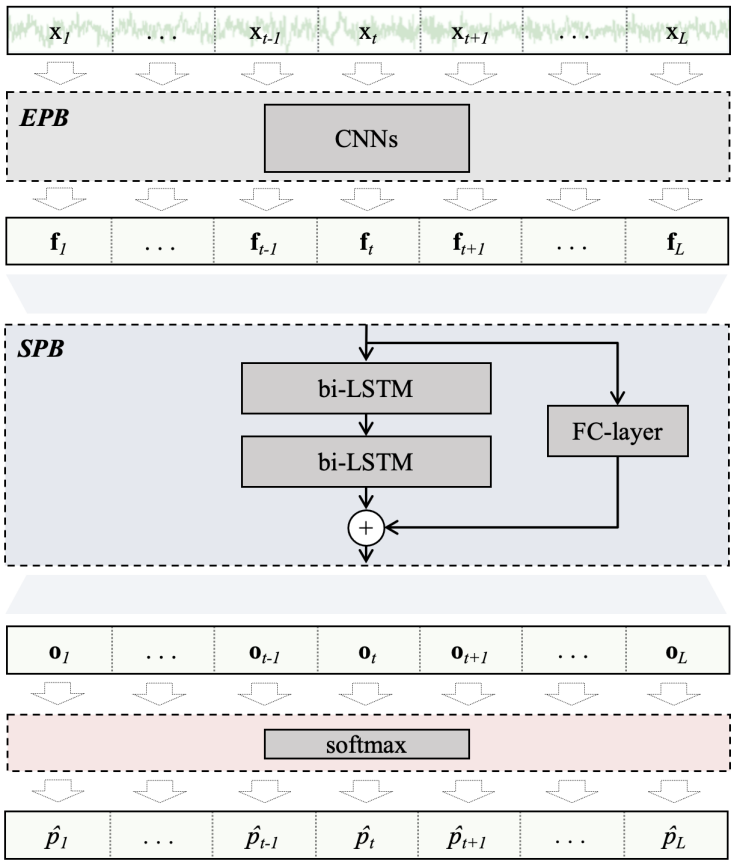


FIGURE B.1: **DeepSleepNet classification scheme.**

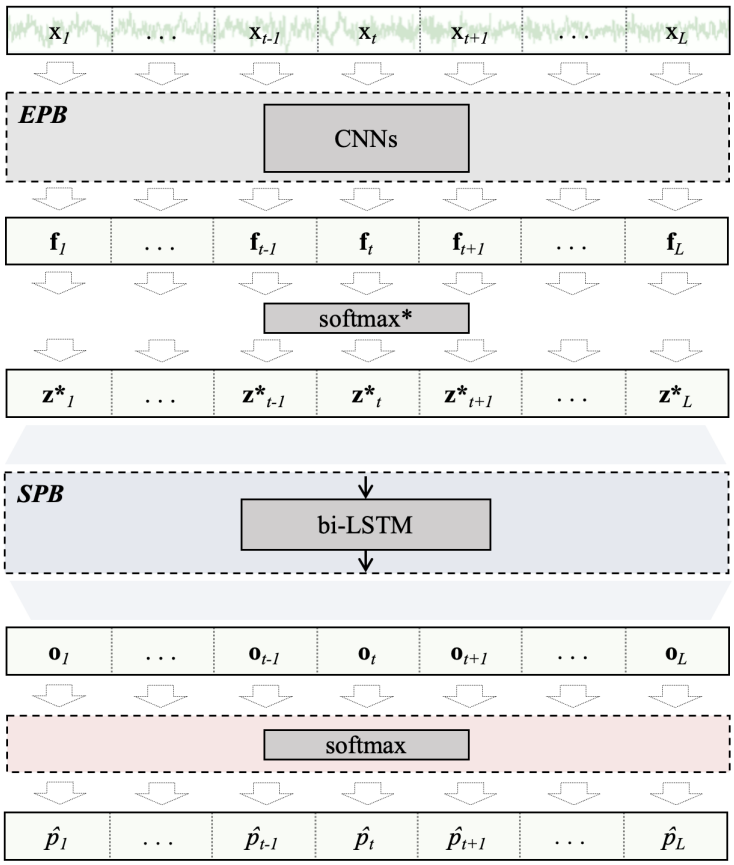


FIGURE B.2: **Sequence-to-sequence bidirectional-LSTM classification scheme.**

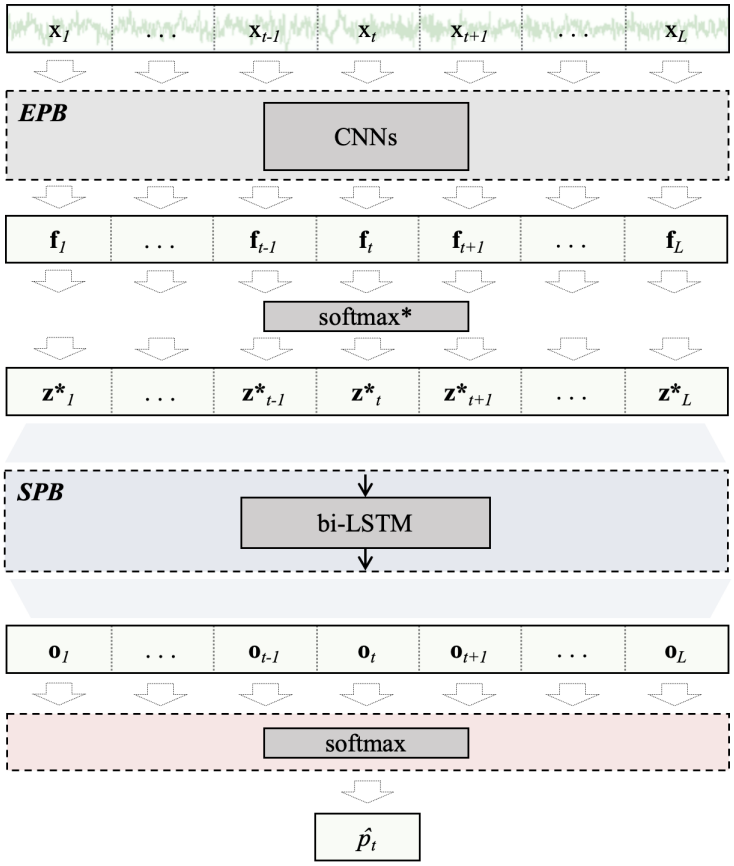


FIGURE B.3: **Sequence-to-epoch bidirectional-LSTM classification scheme.**

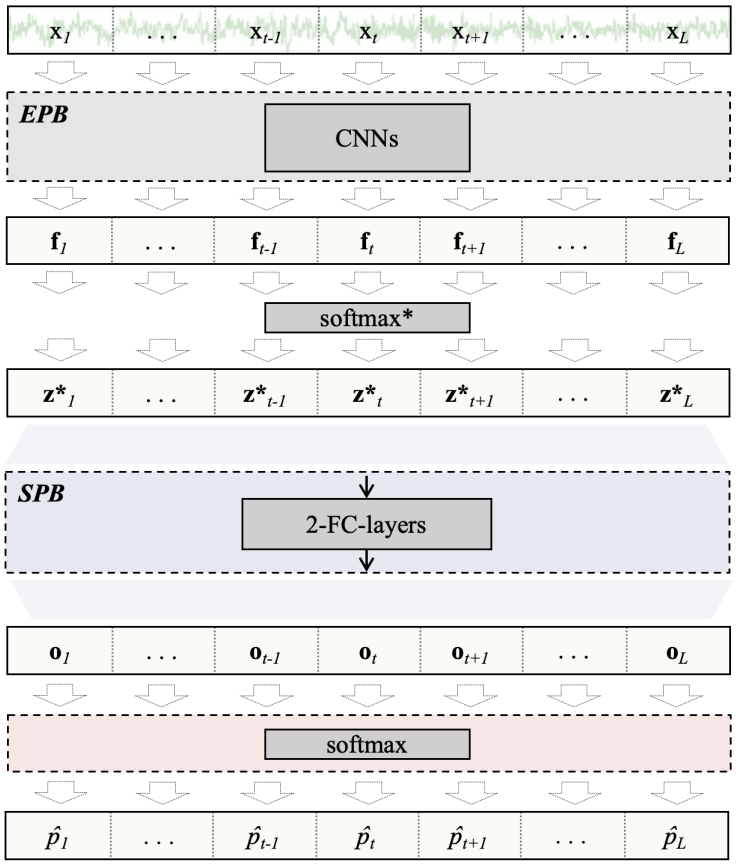


FIGURE B.4: **Sequence-to-sequence FFNN classification scheme.**

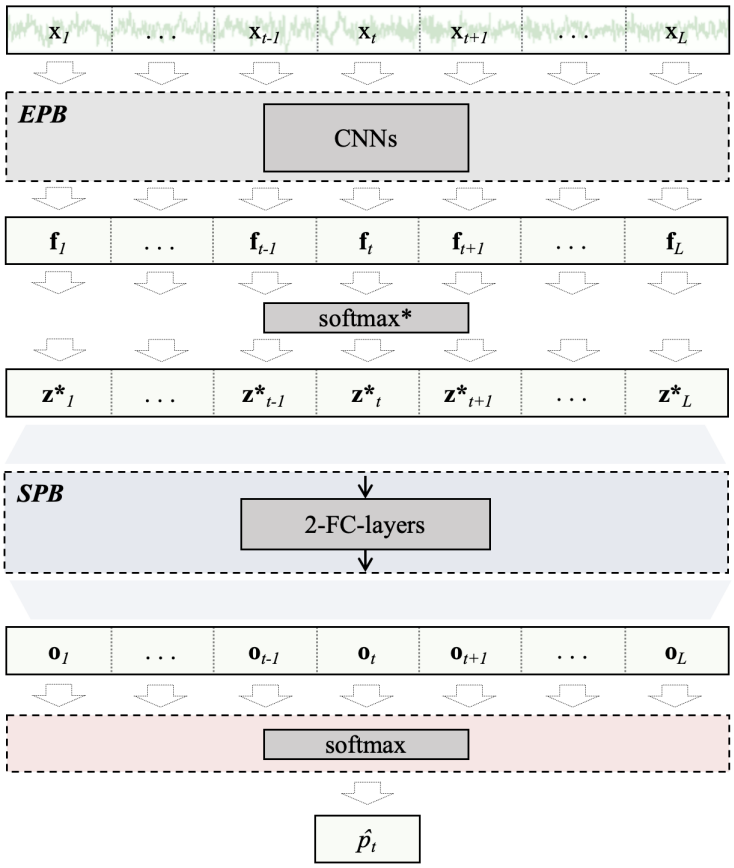


FIGURE B.5: **Sequence-to-epoch FFNN classification scheme.**

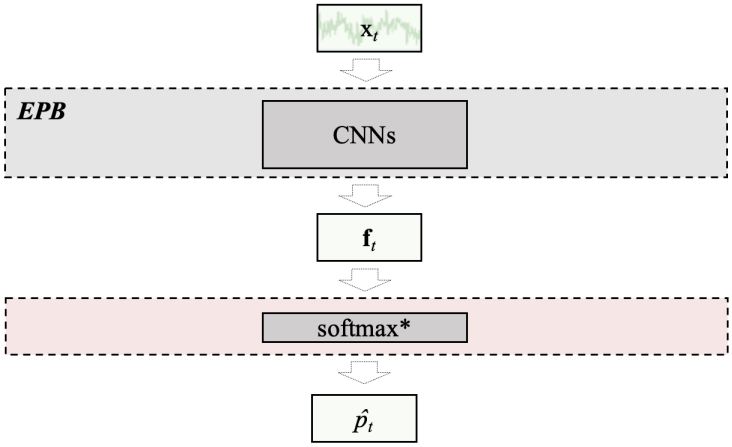


FIGURE B.6: **Epoch-to-epoch EPB classification scheme.**

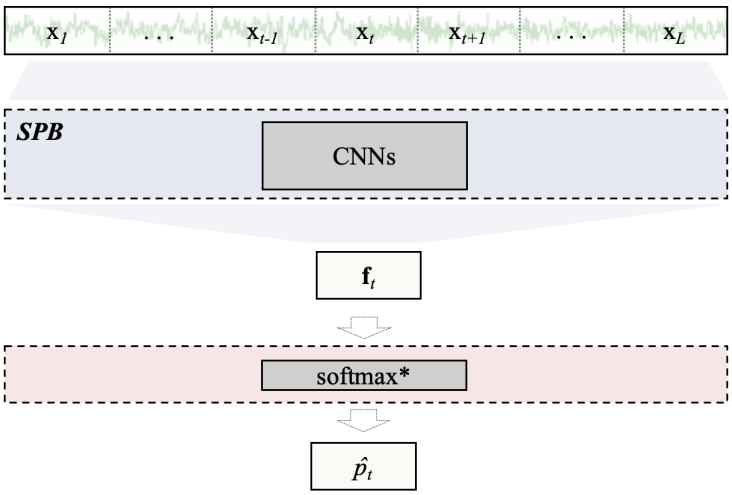


FIGURE B.7: **Sequence-to-epoch SPB classification scheme.**

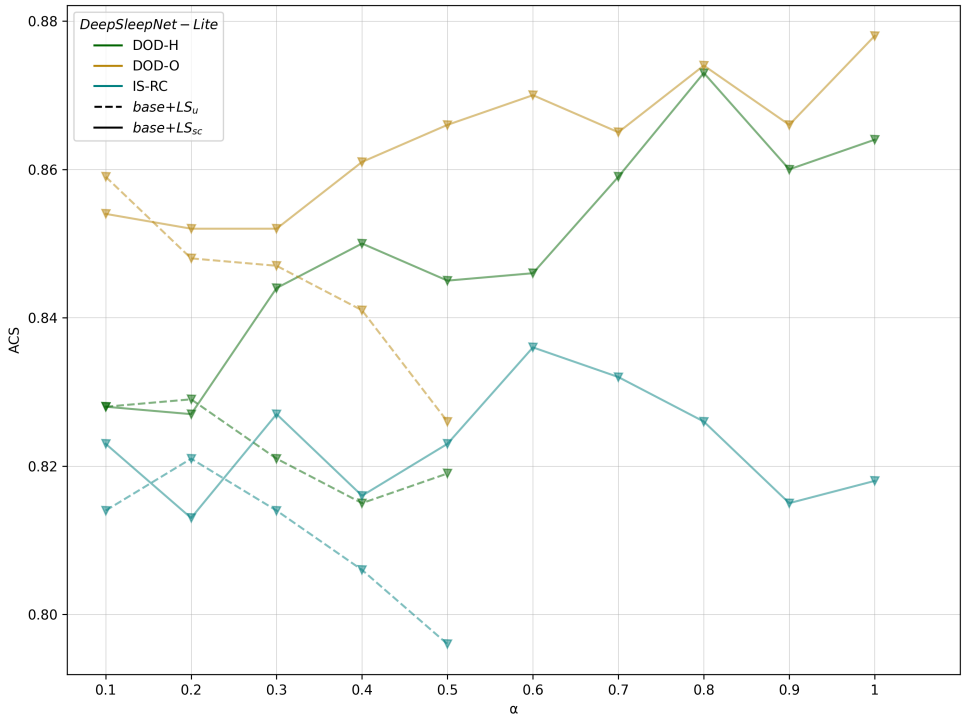


FIGURE B.8: *ACS* **across α values on DSN-L.**
ACS values across all the experimented values, on both the *base*+LS_{U} and the *base*+LS_{SC} DSN-L based models tested on IS-RC, DOD-H and DOD-O datasets.

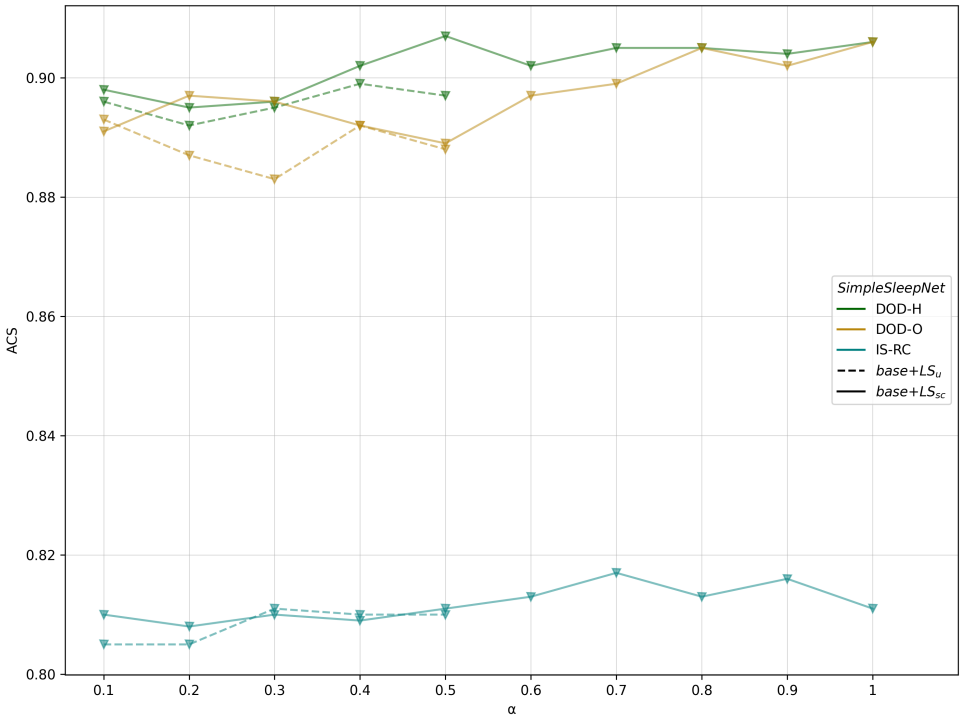


FIGURE B.9: *ACS* **across** α **values on SSN.**
ACS values across all the experimented values, on both the *base+LS*$_{\mathrm{U}}$ and the *base+LS*$_{\mathrm{SC}}$ SSN based models tested on IS-RC, DOD-H and DOD-O datasets.

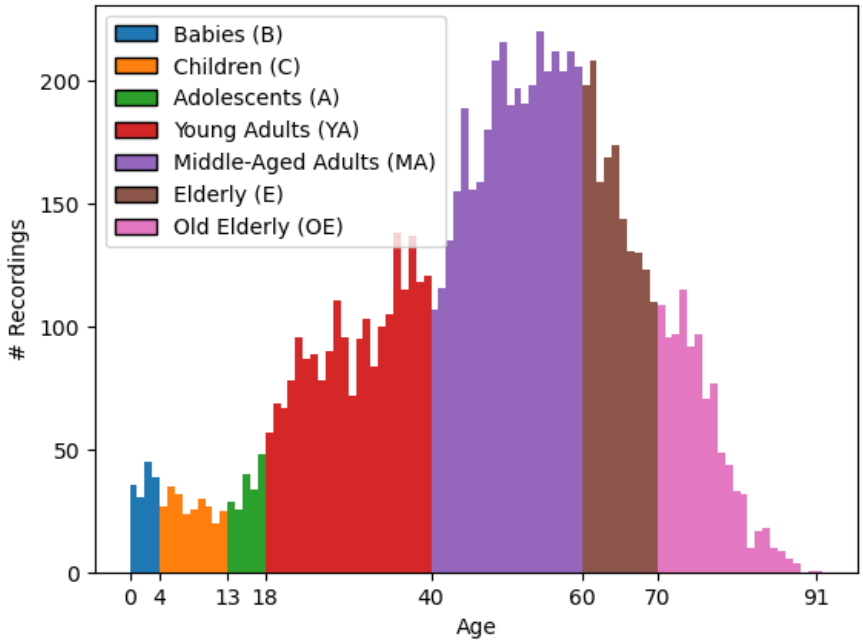


FIGURE B.10: **Age distribution on *BSDB* on seven groups.**
Age distribution on the *BSDB* dataset
in the seven age groups by [105].

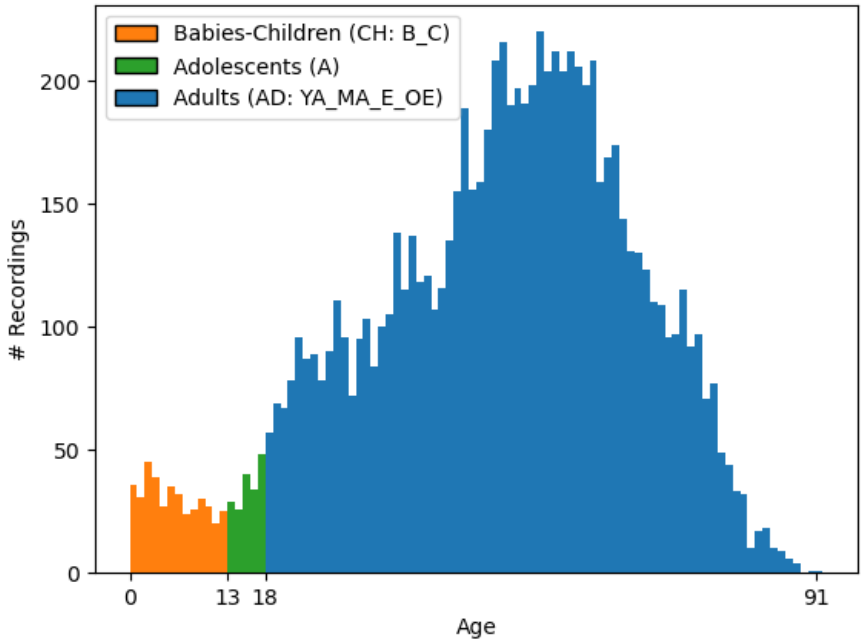


FIGURE B.11: **Age distribution on *BSDB* on three groups.**
Age distribution on the *BSDB* dataset
in the three age groups by AASM [2].

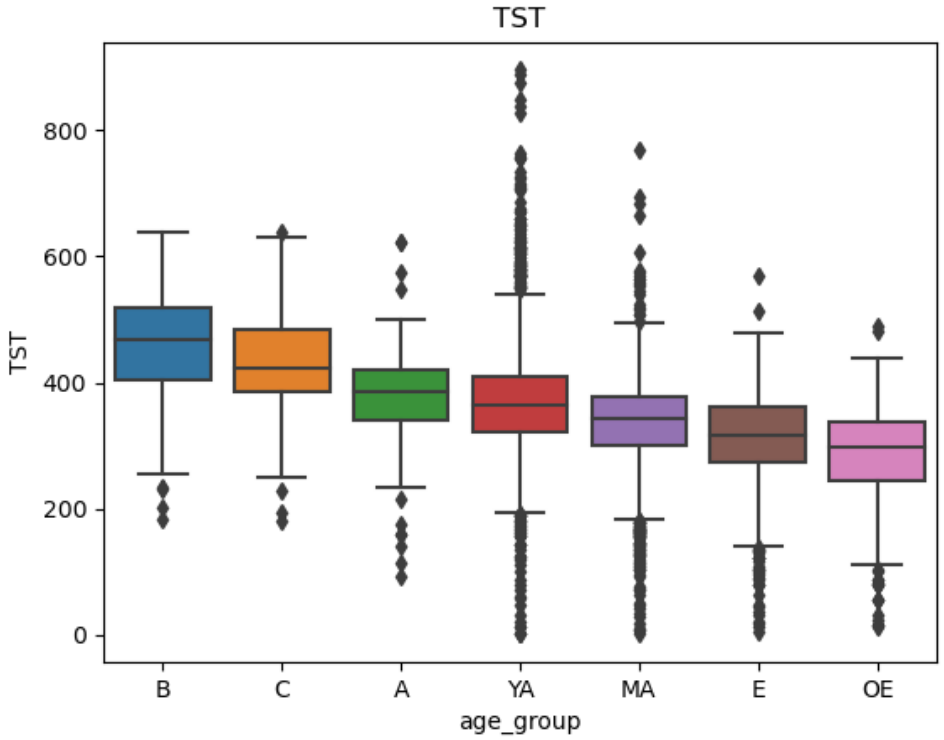


FIGURE B.12: **Boxplots on Total Sleep Time (G=7).**
Boxplots on the Total Sleep Time
for each age group (G=7).

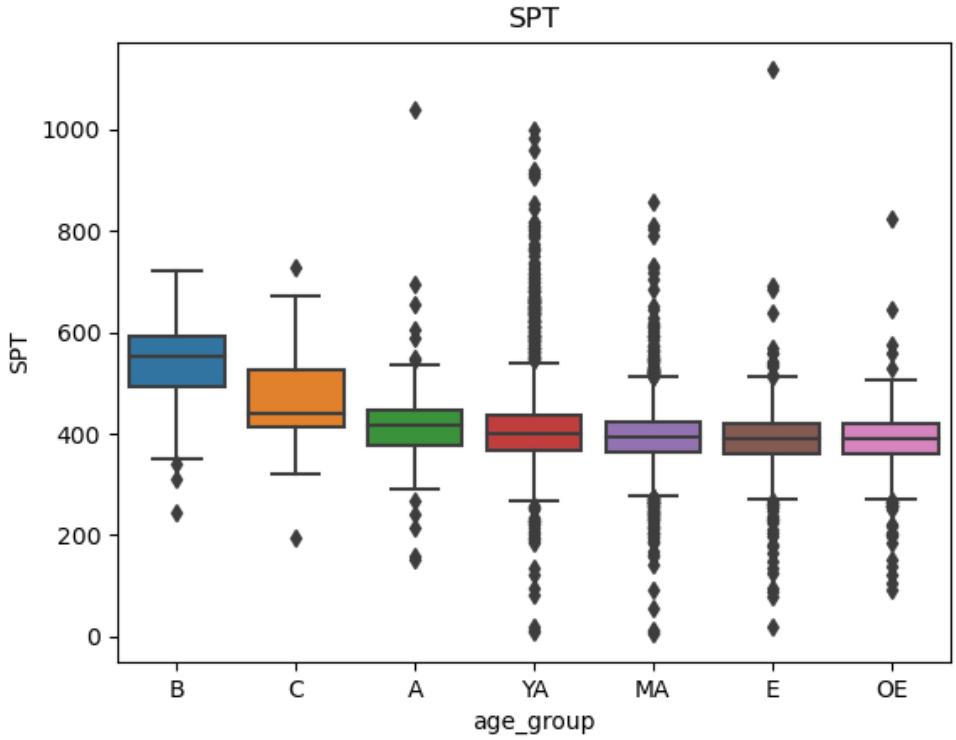


FIGURE B.13: **Boxplots on Sleep Period Time (G=7).**
Boxplots on the Sleep Period Time
for each age group (G=7).

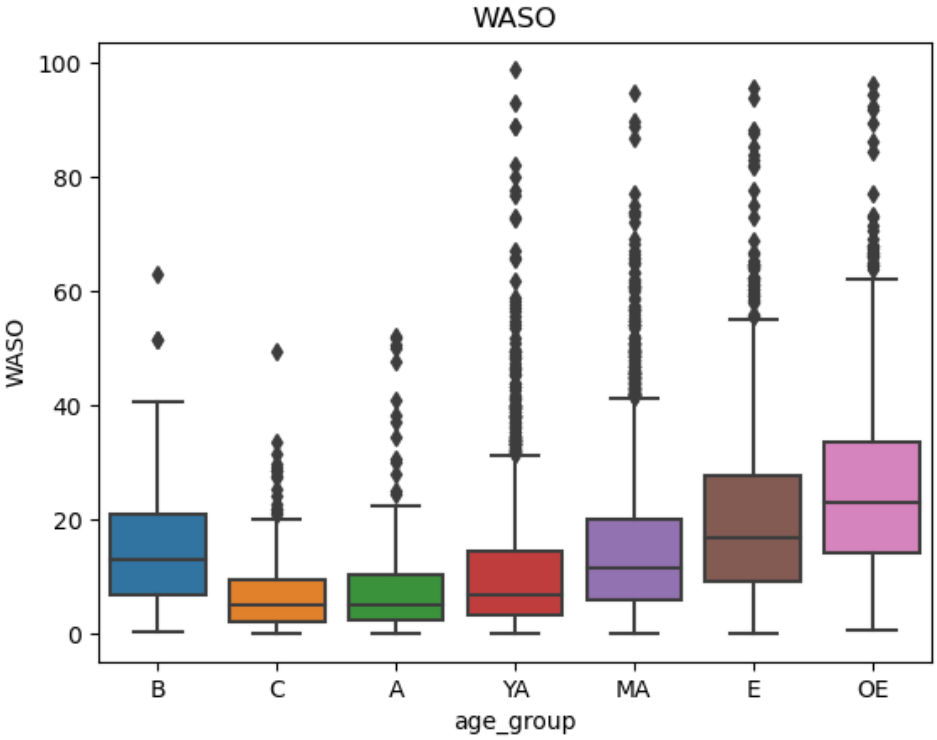


FIGURE B.14: **Boxplots on Wake After Sleep Onset (G=7).**
Boxplots on the Wake After Sleep Onset
for each age group (G=7).

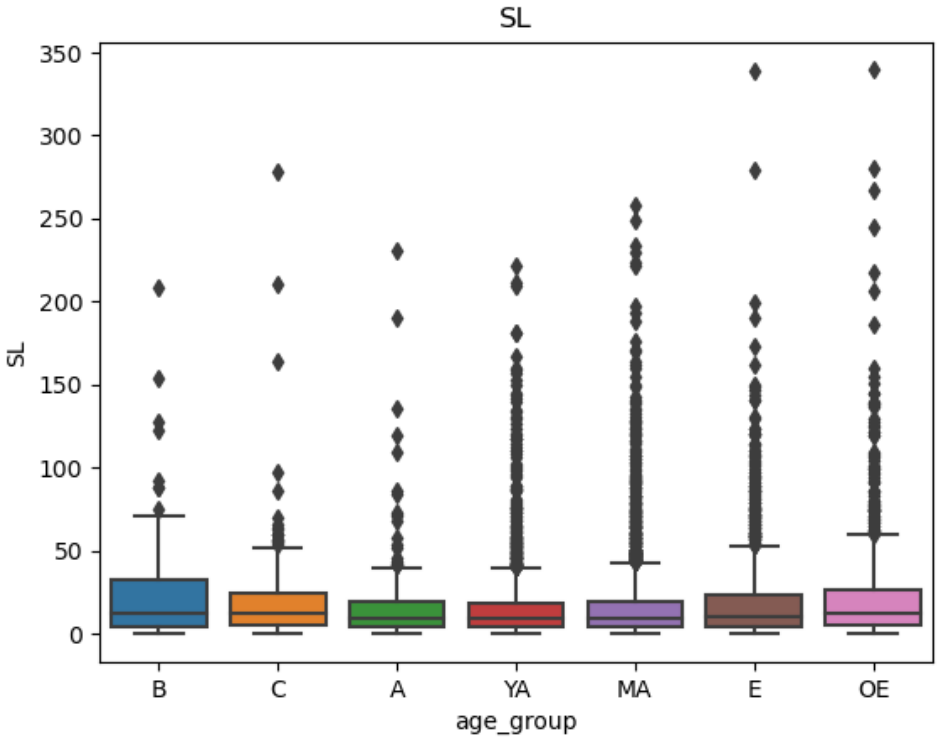


FIGURE B.15: **Boxplots on Sleep Latency (G=7).** Boxplots on the Sleep Latency
for each age group (G=7).

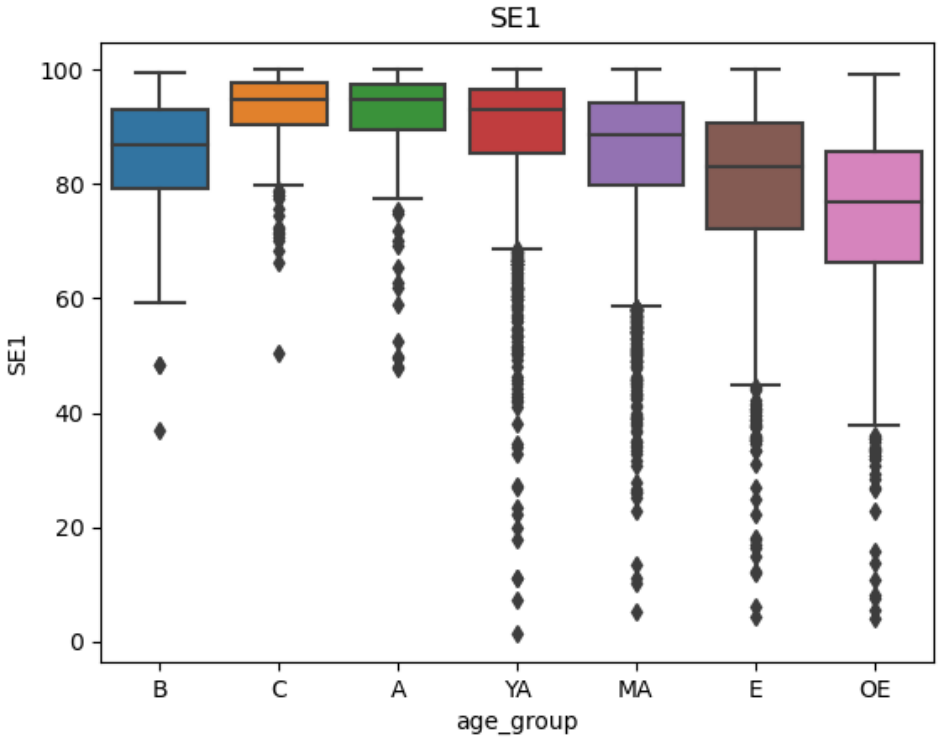


FIGURE B.16: **Boxplots on Sleep Efficiency (G=7).**
Boxplots on the Sleep Efficiency
for each age group (G=7).

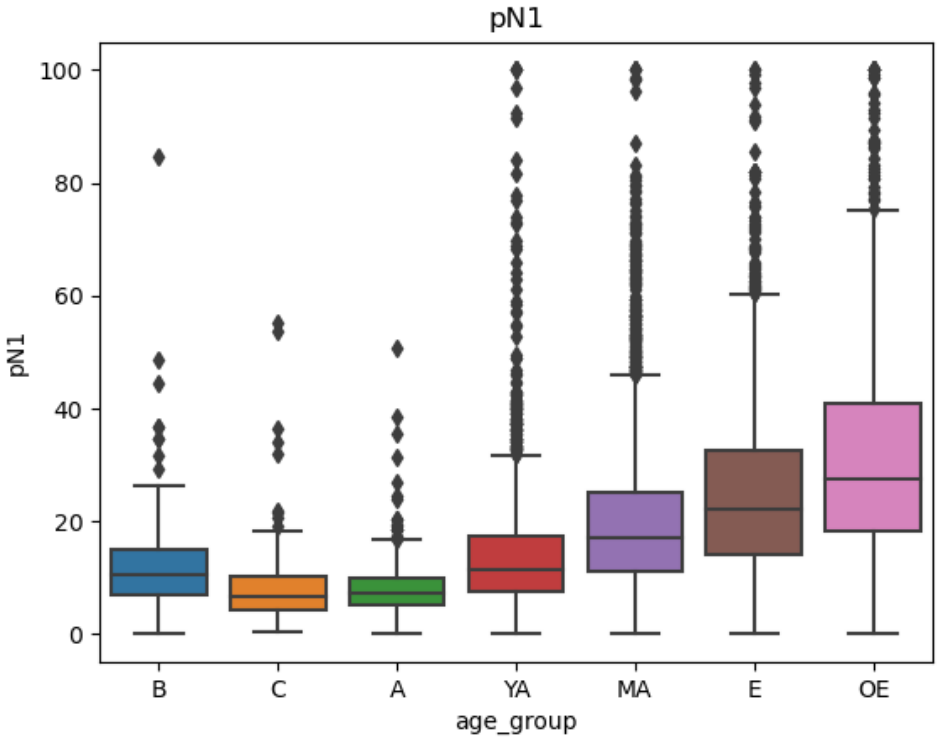


FIGURE B.17: **Boxplots on Percentage of N1 stage (G=7).**
Boxplots on the Percentage of N1 stage
for each age group (G=7).

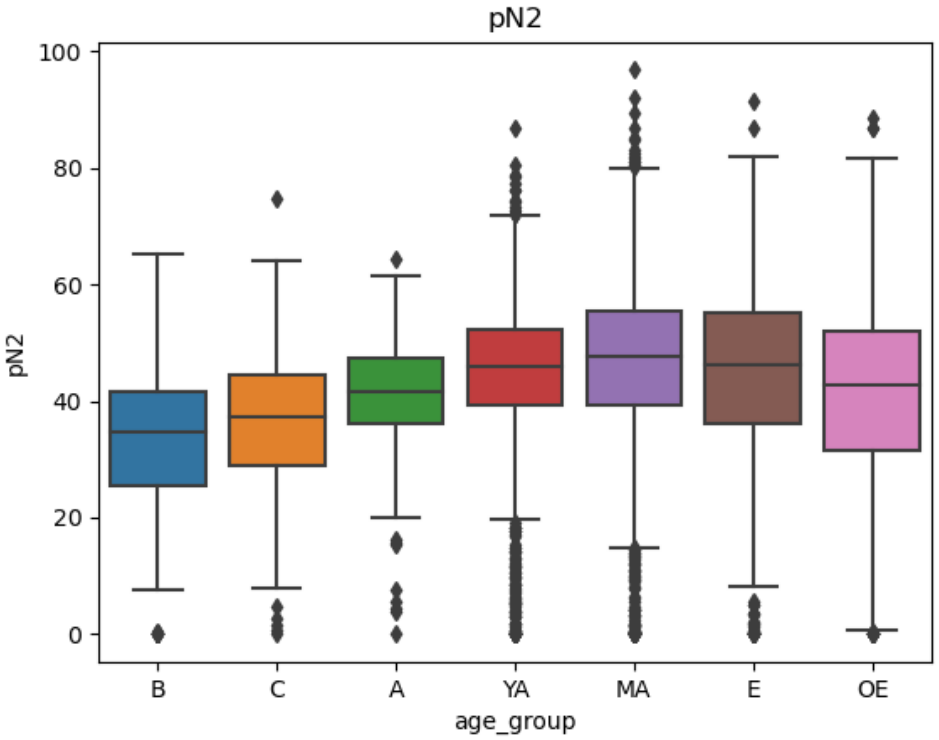


FIGURE B.18: **Boxplots on Percentage of N2 stage (G=7).**
Boxplots on the Percentage of N2 stage
for each age group (G=7).

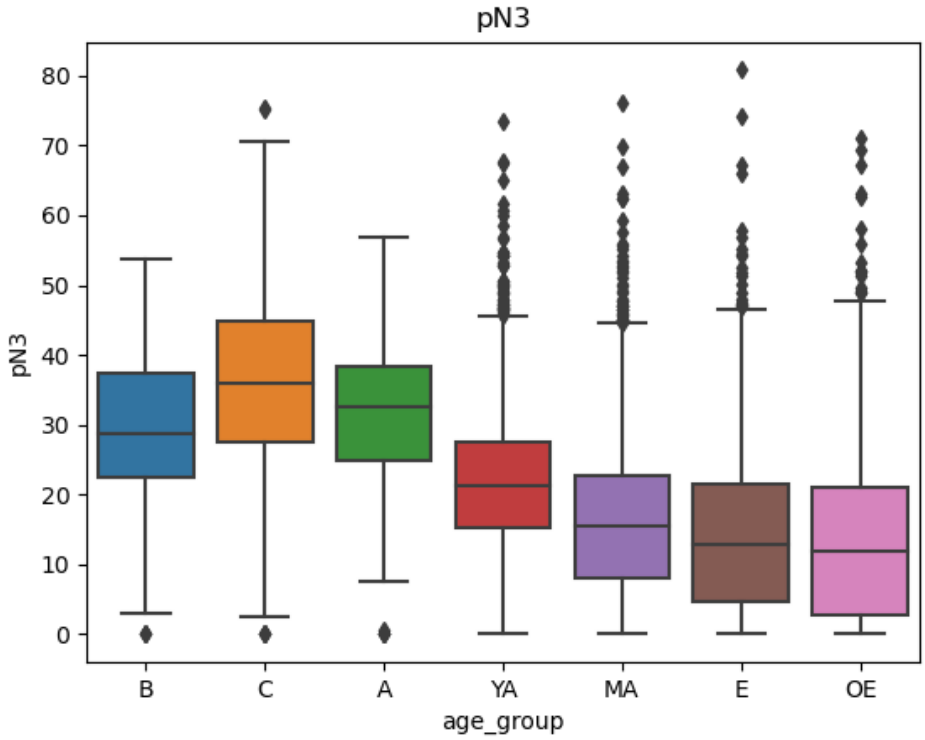


FIGURE B.19: **Boxplots on Percentage of N3 stage (G=7).**
Boxplots on the Percentage of N3 stage
for each age group (G=7).

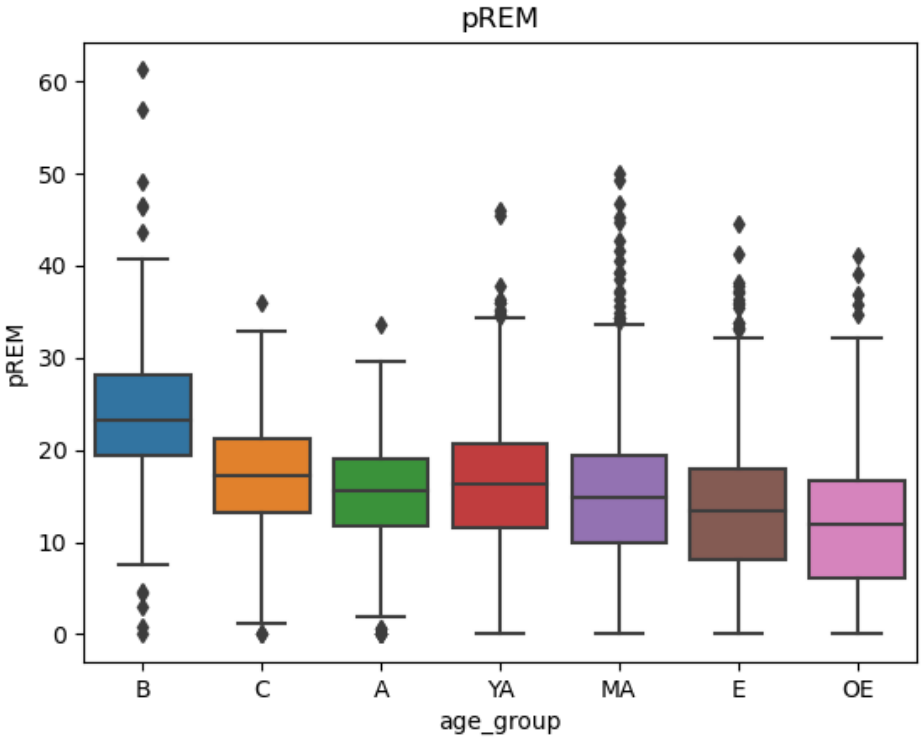


FIGURE B.20: **Boxplots on Percentage of REM stage (G=7)**.
Boxplots on the Percentage of REM stage
for each age group (G=7).

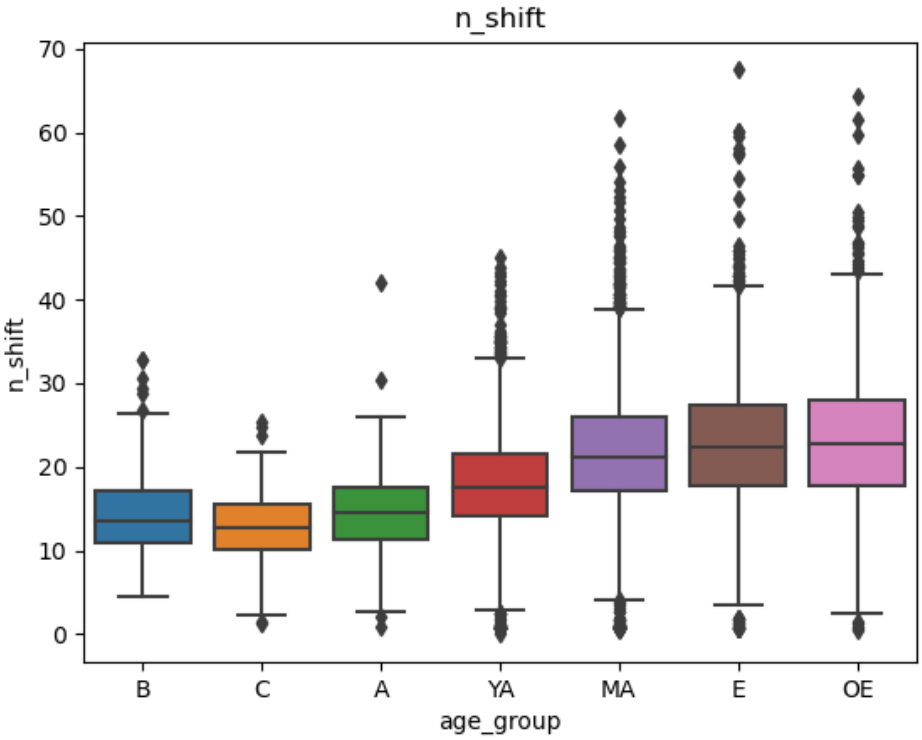


FIGURE B.21: **Boxplots on Number of stage shifts (G=7).**
Boxplots on the Number of stage shifts per hour
for each age group (G=7).

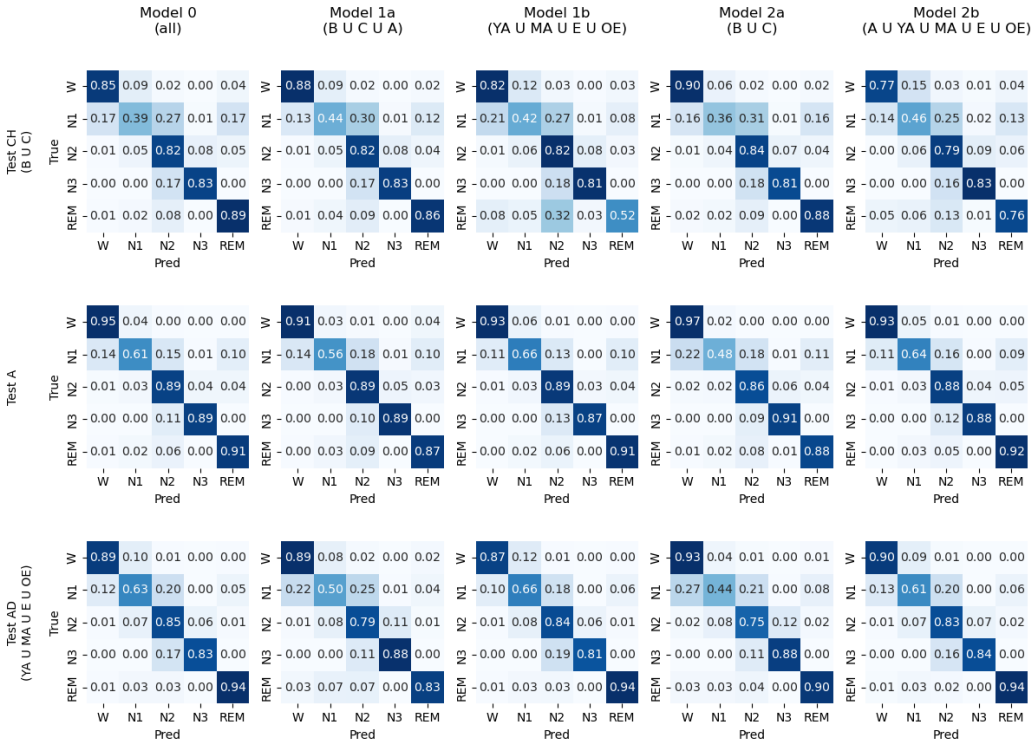


FIGURE B.22: **Confusion matrix on $\{CH, A, AD\}$.**
Confusion matrix of *U-Sleep-v1*
fine-tuned on $\{CH, A, AD\}$.

Bibliography

[1] M. M. Ohayon, "Epidemiological overview of sleep disorders in the general population," *Sleep Medicine Research (SMR)*, vol. 2, no. 1, pp. 1–9, 2011.

[2] C. Iber and C. Iber, *The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications*. American Academy of Sleep Medicine, Westchester, IL, 2007, vol. 1.

[3] R. S. Rosenberg and S. Van Hout, "The american academy of sleep medicine inter-scorer reliability program: Sleep stage scoring," *Journal of clinical sleep medicine*, vol. 9, no. 01, pp. 81–87, 2013.

[4] M. Younes, J. Raneri, and P. Hanly, "Staging sleep in polysomnograms: Analysis of inter-scorer variability," *Journal of Clinical Sleep Medicine*, vol. 12, no. 06, pp. 885–894, 2016.

[5] V Muto, C Berthomier, C Schmidt, *et al.*, "0315 inter-and intra-expert variability in sleep scoring: Comparison between visual and automatic analysis," *Sleep*, vol. 41, no. suppl_1, A121, 2018.

[6] L. Fiorillo, A. Puiatti, M. Papandrea, *et al.*, "Automated sleep scoring: A review of the latest approaches," *Sleep medicine reviews*, vol. 48, p. 101 204, 2019.

[7] L. Fiorillo, M. Wand, I. Marino, P. Favaro, and F. D. Faraci, "Temporal dependency in automatic sleep scoring via deep learning based architectures: An empirical study," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2020, pp. 3509–3512.

[8] L. Fiorillo, P. Favaro, and F. D. Faraci, "Deepsleepnet-lite: A simplified automatic sleep stage scoring model with uncertainty estimates," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 2076–2085, 2021.

[9] A Rechtschaffen and A Kales, *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*, US Government Printing Office, US Public Health Service, 1968.

[10] H. Danker-hopfe, P. Anderer, J. Zeitlhofer, *et al.*, "Interrater reliability for sleep scoring according to the rechtschaffen & kales and the new aasm standard," *Journal of sleep research*, vol. 18, no. 1, pp. 74–84, 2009.

[11] S. Motamedi-Fakhr, M. Moshrefi-Torbati, M. Hill, C. M. Hill, and P. R. White, "Signal processing techniques applied to human sleep eeg signals-a review," *Biomedical Signal Processing and Control*, vol. 10, pp. 21–33, 2014.

[12] B. Şen, M. Peker, A. Çavuşoğlu, and F. V. Çelebi, "A comparative study on classification of sleep stage based on eeg signals using feature selection and classification algorithms," *Journal of medical systems*, vol. 38, no. 3, p. 18, 2014.

[13] K. Aboalayon, M. Faezipour, W. Almuhammadi, and S. Moslehpour, "Sleep stage classification using eeg signal analysis: A comprehensive survey and new investigation," *Entropy*, vol. 18, no. 9, p. 272, 2016.

[14] T. Lan, "Feature extraction feature selection and dimensionality reduction techniques for brain computer interface," Ph.D. dissertation, 2011. [Online]. Available: https://scholararchive.ohsu.edu/concern/etds/np193924b?locale=pt-BR.

[15] M. Ronzhina, O. Janoušek, J. Kolářová, M. Nováková, P. Honzík, and I. Provazník, "Sleep scoring using artificial neural networks," *Sleep medicine reviews*, vol. 16, no. 3, pp. 251–263, 2012.

[16] M. Radha, G. Garcia-Molina, M. Poel, and G. Tononi, "Comparison of feature and classifier algorithms for online automatic sleep staging based on a single eeg signal," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2014, pp. 1876–1880.

[17] R. Boostani, F. Karimzadeh, and M. Nami, "A comparative review on sleep stage classification methods in patients and healthy individuals," *Computer methods and programs in biomedicine*, vol. 140, pp. 77–91, 2017.

[18] S. Güneş, K. Polat, and Ş. Yosunkaya, "Efficient sleep stage recognition system based on eeg signal using k-means clustering based feature weighting," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7922–7928, 2010.

[19] U. R. Acharya, E. C.-P. Chua, K. C. Chua, L. C. Min, and T. Tamura, "Analysis and automatic identification of sleep stages using higher order spectra," *International journal of neural systems*, vol. 20, no. 06, pp. 509–521, 2010.

[20] L. Fraiwan, K Lweesy, N Khasawneh, M Fraiwan, H Wenz, and H Dickhaus, "Classification of sleep stages using multi-wavelet time frequency entropy and lda," *Methods of information in Medicine*, vol. 49, no. 03, pp. 230–237, 2010.

[21] L. Fraiwan, K. Lweesy, N. Khasawneh, H. Wenz, and H. Dickhaus, "Automated sleep stage identification system based on time–frequency analysis of a single eeg channel and random forest classifier," *Computer methods and programs in biomedicine*, vol. 108, no. 1, pp. 10–19, 2012.

[22] S.-F. Liang, C.-E. Kuo, Y.-H. Hu, Y.-H. Pan, and Y.-H. Wang, "Automatic stage scoring of single-channel sleep eeg by using multiscale entropy and autoregressive models," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1649–1657, 2012.

[23] S. Biswal, J. Kulas, H. Sun, *et al.*, "SLEEPNET: automated sleep staging system via deep learning," *CoRR*, vol. abs/1707.08262, 2017. arXiv: 1707.08262. [Online]. Available: http://arxiv.org/abs/1707.08262.

[24] H. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, and Y. Guo, "Mixed neural network approach for temporal sleep stage classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 2, pp. 324–333, 2018.

[25] A. Guillot, F. Sauvet, E. H. During, and V. Thorey, "Dreem open datasets: Multi-scored sleep datasets to compare human and automated sleep staging," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 9, pp. 1955–1965, 2020.

[26] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[28] A. Y. Hannun, C. Case, J. Casper, *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014. arXiv: 1412.5567. [Online]. Available: http://arxiv.org/abs/1412.5567.

[29] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] K. Cho, B. Van Merriënboer, C. Gulcehre, *et al.*, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[32] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. Kamphuisen, and J. J. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the eeg," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, 2000.

[33] C. O'reilly, N. Gosselin, J. Carrier, and T. Nielsen, "Montreal archive of sleep studies: An open-access resource for instrument benchmarking and exploratory research," *Journal of sleep research*, vol. 23, no. 6, pp. 628–635, 2014.

[34] S. F. Quan, B. V. Howard, C. Iber, *et al.*, "The sleep heart health study: Design, rationale, and methods," *Sleep*, vol. 20, no. 12, pp. 1077–1085, 1997.

[35] S. Biswal, H. Sun, B. Goparaju, M. B. Westover, J. Sun, and M. T. Bianchi, "Expert-level sleep scoring with deep neural networks," *Journal of the American Medical Informatics Association*, vol. 25, no. 12, pp. 1643–1650, 2018.

[36] G.-Q. Zhang, L. Cui, R. Mueller, *et al.*, "The national sleep research resource: Towards a sleep data commons," *Journal of the American Medical Informatics Association*, vol. 25, no. 10, pp. 1351–1358, 2018.

[37] J. P. Bakker, A. Tavakkoli, M. Rueschman, *et al.*, "Gastric banding surgery versus continuous positive airway pressure for obstructive sleep apnea: A randomized controlled trial," *American journal of respiratory and critical care medicine*, vol. 197, no. 8, pp. 1080–1083, 2018.

[38] C. L. Rosen, E. K. Larkin, H. L. Kirchner, *et al.*, "Prevalence and risk factors for sleep-disordered breathing in 8-to 11-year-old children: Association with race and prematurity," *The Journal of pediatrics*, vol. 142, no. 4, pp. 383–389, 2003.

[39] S. Redline, P. V. Tishler, T. D. Tosteson, *et al.*, "The familial aggregation of obstructive sleep apnea.," *American journal of respiratory and critical care medicine*, vol. 151, no. 3, pp. 682–687, 1995.

[40] C. L. Marcus, R. H. Moore, C. L. Rosen, *et al.*, "A randomized trial of adenotonsillectomy for childhood sleep apnea," *N Engl J Med*, vol. 368, pp. 2366–2376, 2013.

[41] S. Redline, R. Amin, D. Beebe, *et al.*, "The childhood adenotonsillectomy trial (chat): Rationale, design, and challenges of a randomized controlled trial evaluating a standard surgical procedure in a pediatric population," *Sleep*, vol. 34, no. 11, pp. 1509–1517, 2011.

[42] V. Thorey, A. B. Hernandez, P. J. Arnal, and E. H. During, "Ai vs humans for the diagnosis of sleep apnea," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2019, pp. 1596–1600.

[43] S. Khalighi, T. Sousa, J. M. Santos, and U. Nunes, "Isruc-sleep: A comprehensive public dataset for sleep researchers," *Computer methods and programs in biomedicine*, vol. 124, pp. 180–192, 2016.

[44] C. L. Rosen, D. Auckley, R. Benca, *et al.*, "A multisite randomized trial of portable sleep studies and positive airway pressure autotitration versus laboratory-based polysomnography for the diagnosis and treatment of obstructive sleep apnea: The homepap study," *Sleep*, vol. 35, no. 6, pp. 757–767, 2012.

[45] X. Chen, R. Wang, P. Zee, *et al.*, "Racial/ethnic differences in sleep disturbances: The multi-ethnic study of atherosclerosis (mesa)," *Sleep*, vol. 38, no. 6, pp. 877–888, 2015.

[46] T. Blackwell, K. Yaffe, S. Ancoli-Israel, *et al.*, "Associations between sleep architecture and sleep-disordered breathing and cognition in older community-dwelling men: The osteoporotic fractures in men sleep study," *Journal of the American Geriatrics Society*, vol. 59, no. 12, pp. 2217–2225, 2011.

[47] "Relationships between sleep stages and changes in cognitive function in older men: The mros sleep study," *Sleep*, vol. 38, no. 3, pp. 411–421, 2015.

[48] A. L. Goldberger, L. A. Amaral, L. Glass, *et al.*, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, e215–e220, 2000.

[49] M. M. Ghassemi, B. E. Moody, L.-W. H. Lehman, *et al.*, "You snooze, you win: The physionet/computing in cardiology challenge 2018," in *2018 Computing in Cardiology Conference (CinC)*, IEEE, vol. 45, 2018, pp. 1–4.

[50] S. R. Cummings, D. M. Black, M. C. Nevitt, *et al.*, "Appendicular bone density and age predict hip fracture in women," *Jama*, vol. 263, no. 5, pp. 665–668, 1990.

[51] A. P. Spira, T. Blackwell, K. L. Stone, *et al.*, "Sleep-disordered breathing and cognition in older women," *Journal of the American Geriatrics Society*, vol. 56, no. 1, pp. 45–50, 2008.

[52] A. Patanaik, J. L. Ong, J. J. Gooley, S. Ancoli-Israel, and M. W. Chee, "An end-to-end framework for real-time automatic sleep stage classification," *Sleep*, vol. 41, no. 5, zsy041, 2018.

[53] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 4, pp. 758–769, 2018.

[54] A. Malafeev, D. Laptev, S. Bauer, *et al.*, "Automatic human sleep stage scoring using deep neural networks," *Frontiers in Neuroscience*, vol. 12, p. 781, 2018.

[55] O. Tsinalis, P. M. Matthews, and Y. Guo, "Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders," *Annals of biomedical engineering*, vol. 44, no. 5, pp. 1587–1597, 2016.

[56] M. Perslev, M. Jensen, S. Darkner, P. J. Jennum, and C. Igel, "U-time: A fully convolutional network for time series segmentation applied to sleep staging," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[57] M. Perslev, S. Darkner, L. Kempfner, M. Nikolic, P. J. Jennum, and C. Igel, "U-sleep: Resilient high-frequency sleep staging," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–12, 2021.

[58] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, "Automatic sleep stage scoring with single-channel eeg using convolutional neural networks," *arXiv preprint arXiv:1610.01683*, 2016.

[59] A. Vilamala, K. H. Madsen, and L. K. Hansen, "Deep convolutional neural networks for interpretable analysis of eeg sleep stage scoring," in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2017, pp. 1–6.

[60] J. Zhang and Y. Wu, "Complex-valued unsupervised convolutional neural networks for sleep stage classification," *Computer methods and programs in biomedicine*, vol. 164, pp. 181–191, 2018.

[61] Z. Cui, X. Zheng, X. Shao, and L. Cui, "Automatic sleep stage classification based on convolutional neural network and fine-grained segments," *Complexity*, vol. 2018, 2018.

[62] A. N. Olesen, P. Jennum, P. Peppard, E. Mignot, and H. B. Sorensen, "Deep residual networks for automatic sleep stage classification of raw polysomnographic waveforms," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2018, pp. 1–4.

[63] A. Sors, S. Bonnet, S. Mirek, L. Vercueil, and J.-F. Payen, "A convolutional neural network for sleep stage scoring from raw single-channel eeg," *Biomedical Signal Processing and Control*, vol. 42, pp. 107–114, 2018.

[64] O. Yildirim, U. B. Baloglu, and U. R. Acharya, "A deep learning model for automated sleep stages classification using psg signals," *International journal of environmental research and public health*, vol. 16, no. 4, p. 599, 2019.

[65] N. Michielli, U. R. Acharya, and F. Molinari, "Cascaded lstm recurrent neural network for automated sleep stage classification using single-channel eeg signals," *Computers in biology and medicine*, vol. 106, pp. 71–81, 2019.

[66] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Seqsleepnet: End-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, p. 1, 2019, ISSN: 1534-4320. DOI: 10.1109/TNSRE.2019.2896659.

[67] J. B. Stephansen, A. N. Olesen, M. Olsen, *et al.*, "Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy," *Nature communications*, vol. 9, no. 1, p. 5229, 2018.

[68] S. Back, S. Lee, H. Seo, D. Park, T. Kim, and K. Lee, "Intra-and inter-epoch temporal context network (iitnet) for automatic sleep stage scoring," *arXiv preprint arXiv:1902.06562*, 2019.

[69] S. Mousavi, F. Afghah, and U. R. Acharya, "Sleepeegnet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PloS one*, vol. 14, no. 5, e0216456, 2019.

[70] H. Seo, S. Back, S. Lee, D. Park, T. Kim, and K. Lee, "Intra-and inter-epoch temporal context network (iitnet) using sub-epoch features for automatic sleep scoring on raw single-channel eeg," *Biomedical Signal Processing and Control*, vol. 61, p. 102 037, 2020.

[71] H. Phan, O. Y. Chén, P. Koch, *et al.*, "Towards more accurate automatic sleep staging via deep transfer learning," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 6, pp. 1787–1798, 2020.

[72] A. Supratak and Y. Guo, "Tinysleepnet: An efficient deep learning model for sleep stage scoring based on raw single-channel eeg," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2020, pp. 641–644.

[73] H. Phan, O. Y. Chén, M. C. Tran, P. Koch, A. Mertins, and M. De Vos, "Xsleepnet: Multi-view sequential model for automatic sleep staging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[74] R. G. Fichman, R. Kohli, and R. Krishnan, "Editorial overview-the role of information systems in healthcare: Current research and future trends," *Information Systems Research*, vol. 22, no. 3, pp. 419–428, 2011.

[75] R. Marcilly, L. Peute, and M.-C. Beuscart-Zephir, "From usability engineering to evidence-based usability in health it," *Stud Health Technol Inform*, vol. 222, pp. 126–38, 2016.

[76] A. Kushniruk and C. Nøhr, "Participatory design, user involvement and health it evaluation," *Stud Health Technol Inform*, vol. 222, pp. 139–151, 2016.

[77] J. Tay, S. Toh, L. Leow, and S. Senin, "Assessing competency of z3score automated sleep stage scoring system with manual sleep stage scoring by multiple scorers," *Sleep Medicine*, vol. 40, e326, 2017.

[78] M. Younes, W. Thompson, C. Leslie, T. Egan, and E. Giannouli, "Utility of technologist editing of polysomnography scoring performed by a validated automatic system," *Annals of the American Thoracic Society*, vol. 12, no. 8, pp. 1206–1218, 2015.

[79] O. Ali, A. Shrestha, J. Soar, and S. F. Wamba, "Cloud computing-enabled healthcare opportunities, issues, and applications: A systematic review," *International Journal of Information Management*, vol. 43, pp. 146–158, 2018.

[80] P. S. Jensen, H. B. Sorensen, H. L. Leonthin, and P. Jennum, "Automatic sleep scoring in normals and in individuals with neurodegenerative disorders according to new international sleep scoring criteria," *Journal of Clinical Neurophysiology*, vol. 27, no. 4, pp. 296–302, 2010.

[81] R. K. Malhotra and A. Y. Avidan, "Introduction to sleep stage scoring," *Atlas of Sleep Medicine*, p. 77, 2013.

[82] M. Younes, S. T. Kuna, A. I. Pack, *et al.*, "Reliability of the american academy of sleep medicine rules for assessing sleep depth in clinical practice," *Journal of Clinical Sleep Medicine*, vol. 14, no. 02, pp. 205–213, 2018.

[83] F. Andreotti, H. Phan, N. Cooray, C. Lo, M. T. Hu, and M. De Vos, "Multi-channel sleep stage classification and transfer learning using convolutional neural networks," in *2018 40th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2018, pp. 171–174.

[84] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Joint classification and prediction cnn framework for automatic sleep stage classification," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 5, pp. 1285–1296, 2018.

[85] M. X. Cohen, *Analyzing neural time series data: theory and practice*. MIT press, 2014.

[86] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, PMLR, 2015, pp. 448–456.

[87] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[88] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[89] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[90] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[91] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[92] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[93] L. Prechelt, "Early stopping-but when?" In *Neural Networks: Tricks of the trade*, Springer, 1998, pp. 55–69.

[94] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[95] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[96] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *Advances in neural information processing systems*, vol. 32, 2019.

[97] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, PMLR, 2016, pp. 1050–1059.

[98] S. A. Imtiaz and E. Rodriguez-Villegas, "An open-source toolbox for standardized use of physionet sleep edf expanded database," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2015, pp. 6014–6017.

[99] S. T. Kuna, R. Benca, C. A. Kushida, *et al.*, "Agreement in computer-assisted manual scoring of polysomnograms across sleep centers," *Sleep*, vol. 36, no. 4, pp. 583–589, 2013.

[100] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[101] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[102] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*, PMLR, 2017, pp. 1321–1330.

[103] Y. Nir, R. Staba, T. Andrillon, *et al.*, "Regional slow waves and spindles in human sleep," *Neuron*, vol. 70, no. 1, pp. 153–169, 2011.

[104] F. Siclari, C. Bassetti, and G. Tononi, "Conscious experience in sleep and wakefulness," *Swiss Arch Neurol Psychiatry*, vol. 163, pp. 273–8, 2012.

[105] M. Ohayon, M. Carskadon, C. Guilleminault, and M. Vitiello, "Meta-analysis of quantitative sleep parameters from childhood to old age in healthy individuals: Developing normative sleep values across the human lifespan," *Sleep*, vol. 27, pp. 1255–73, Dec. 2004. DOI: 10.1093/sleep/27.7.1255.

[106] D. Kocevska, T. S. Lysen, A. Dotinga, *et al.*, "Sleep characteristics across the lifespan in 1.1 million people from the netherlands, united kingdom and united states: A systematic review and meta-analysis," *Nature human behaviour*, vol. 5, no. 1, pp. 113–122, 2021.

[107] A. Guillot and V. Thorey, "Robustsleepnet: Transfer learning for automated sleep staging at scale," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 1441–1451, 2021.

[108] A. N. Olesen, P. Jørgen Jennum, E. Mignot, and H. B. D. Sorensen, "Automatic sleep stage classification with deep residual networks in a mixed-cohort setting," *Sleep*, vol. 44, no. 1, zsaa161, 2021.

[109] R. Vallat and M. P. Walker, "An open-source, high-performance tool for automated sleep staging," *Elife*, vol. 10, e70092, 2021.

[110] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[111] T. Falk, D. Mai, R. Bensch, *et al.*, "U-net: Deep learning for cell counting, detection, and morphometry," *Nature methods*, vol. 16, no. 1, pp. 67–70, 2019.

[112] M. Brandt, C. J. Tucker, A. Kariryaa, *et al.*, "An unexpectedly large count of trees in the west african sahara and sahel," *Nature*, vol. 587, no. 7832, pp. 78–82, 2020.

[113] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[114] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, Elsevier, 1989, pp. 109–165.

[115] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016.

[116] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[117] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[118] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[119] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 819–828.

[120] X. Gong, W. Chen, T. Chen, and Z. Wang, "Sandwich batch normalization: A drop-in replacement for feature distribution heterogeneity," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2494–2504.

[121] J. Mathis, D. Andres, W. Schmitt, C. Bassetti, C. Hess, and D. Schreier, "The diagnostic value of sleep and vigilance tests in central disorders of hypersomnolence," *Sleep*, vol. 45(3), 2022.

[122] M. M. Grigg-Damberger, "The visual scoring of sleep in infants 0 to 2 months of age," *Journal of clinical sleep medicine*, vol. 12, no. 3, pp. 429–445, 2016.

[123] H. Lee, B. Li, S. DeForte, *et al.*, "Nch sleep databank: A large collection of real-world pediatric sleep studies," *arXiv preprint arXiv:2102.13284*, 2021.

[124] J. Frohlich, L. M. Bird, J. Dell'Italia, M. A. Johnson, J. F. Hipp, and M. M. Monti, "High-voltage, diffuse delta rhythms coincide with wakeful consciousness and complexity in angelman syndrome," *Neuroscience of consciousness*, vol. 2020, no. 1, niaa005, 2020.

[125] J. Frohlich, J. N. Chiang, P. A. Mediano, *et al.*, "Neural complexity is a common denominator of human consciousness across diverse regimes of cortical dynamics,"

[126] A. Hyvarinen, "Fast ica for noisy data using gaussian moments," in *1999 IEEE international symposium on circuits and systems (ISCAS)*, IEEE, vol. 5, 1999, pp. 57–61.

Declaration of consent

on the basis of Article 18 of the PromR Phil.-nat. 19

Name/First Name:

Registration Number:

Study program:

Bachelor ☐ Master ☐ Dissertation ☐

Title of the thesis:

Supervisor:

I declare herewith that this thesis is my own work and that I have not used any sources other than those stated. I have indicated the adoption of quotations as well as thoughts taken from other authors as such in the thesis. I am aware that the Senate pursuant to Article 36 paragraph 1 litera r of the University Act of September 5th, 1996 and Article 69 of the University Statute of June 7th, 2011 is authorized to revoke the doctoral degree awarded on the basis of this thesis.
For the purposes of evaluation and verification of compliance with the declaration of originality and the regulations governing plagiarism, I hereby grant the University of Bern the right to process my personal data and to perform the acts of use this requires, in particular, to reproduce the written thesis and to store it permanently in a database, and to use said database, or to make said database available, to enable comparison with theses submitted by others.

Place/Date

Signature