A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series

Stanislas Chambon*[1, 2], Mathieu N. Galtier†[1], Pierrick J. Arnal‡[1], Gilles Wainrib§[3, 4], and Alexandre Gramfort¶[5, 6]

[1]Research & Algorithms Team, Rythm inc., Paris, France
[2]LTCI, Telecom ParisTech, Université Paris-Saclay, Paris, France
[3]DATA Team, Département d'Informatique, École Normale Supérieure, Paris, France
[4]Owkin inc., Paris, France
[5]INRIA, Université Paris-Saclay, Paris, France
[6]CEA, Université Paris-Saclay, Paris, France

July 12, 2017

Abstract

Objective: Sleep stage classification constitutes an important preliminary exam in the diagnosis of sleep disorders and is traditionally performed by a sleep expert who assigns to each 30 s of signal a sleep stage, based on the visual inspection of signals such as electroencephalograms (EEG), electrooculograms (EOG), electrocardiograms (ECG) and electromyograms (EMG). In this paper, we introduce the first end-to-end deep learning approach that performs automatic temporal sleep stage classification from multivariate and multimodal Polysomnography (PSG) signals. *Methods*: We build a general deep architecture which can extract information from EEG, EOG and EMG channels and pools the learnt representations into a final softmax classifier. For each modality the first layer learns linear spatial filters that exploit the array of sensors to increase the signal-to-noise ratio. The architecture is light enough to be distributed in time in order to learn from the temporal context of each sample, namely previous and following data segments. *Results*: Our model, which is unique in its ability to learn a feature representation from multiple modalities, is compared to alternative automatic approaches based on convolutional networks or decisions trees. Results obtained on 61 publicly available PSG records with up to 20 EEG channels demonstrate that our network architecture yields state-of-the-art performance. *Conclusion*: Our study reveals a number of insights on the spatio-temporal distribution of the signal of interest: a good trade-off for optimal classification performance measured with balanced accuracy is to use 6 EEG with some EOG and EMG channels. Also exploiting one minute of data before and after each data segment to be classified offers the strongest improvement when a limited number of channels is available. *Significance*: Our approach aims to improve a key step in the study of sleep disorders. As sleep experts, our system exploits the multivariate and multimodal character of PSG signals to deliver state-of-the-art classification performance at a very low complexity cost.

Keywords: Sleep stage classification, multivariate time series, deep learning, spatio-temporal data, transfer learning, EEG, EOG, EMG

1 Introduction

Sleep stage identification, *a.k.a. sleep scoring* or *sleep stage classification*, is of great interest to better understand sleep and its disorders. Indeed, the construction of an hypnogram, the sequence of sleep stages over a night, is often involved, as a preliminary exam, in the diagnosis of sleep disorders such as insomnia or sleep apnea [6]. Traditionally, this exam is performed as follows. First a subject sleeps with a medical device which performs a polysomnography (PSG), *i.e.*, it records electroencephalography (EEG) signals at different locations over the head, electrooculography (EOG) signals, electromyography (EMG) signals, and eventually more. Second, a human sleep expert looks at the different time series recorded over the night and assigns to each 30 s time

*stanislas@rythm.co
†mathieu@rythm.co
‡pierrick@rythm.co
§gilles.wainrib@ens.fr
¶alexandre.gramfort@inria.fr

segment a sleep stage following a reference nomenclature such as American Academy of Sleep Medicine (AASM) rules [18] or Rechtschaffen and Kales (RK) rules [2]. Regarding the AASM rules, 5 stages are identified: Wake (W), Rapid Eye Movements (REM), Non REM1 (N1), Non REM2 (N2) and Non REM3 (N3) also known as slow wave sleep or even deep sleep. They are characterized by distinct time and frequency patterns and they also differ in proportions over a night. For instance, transitory stages such as N1 are less frequent than REM or N2. In the case of AASM rules, the transitions between two different stages are also documented and the transition rules may modulate the final decision of a human scorer. Indeed, some transitions are prohibited or others are strengthened depending on the occurence of some events such as arousal, K-complex or spindles regarding the transition N1-N2 [18, 35]. Although very precious information is collected thanks to this exam, sleep scoring is a tedious and time consuming task which is furthermore subject to the scorer subjectivity and variability [28].

The use of automatic sleep scoring methods or at least an automatic assistance has been investigated for several years and has driven much interest. From a statistical machine learning perspective, the problem is an imbalanced multi-class prediction problem. State-of-the-art automatic approaches can be classified into two categories depending on whether the features used for classification are extracted using expert knowledge or if they are learnt from the raw signals. Methods of the first category rely on a priori knowledge about the signals and events that enables to design hand-crafted features (see [1] for a very extensive list of references). Methods in the second category consist in learning appropriate feature representations from raw data with convolutional neural networks [36, 33].

One of the main statistical learning challenges is the imbalanced nature of the classification task which has important practical implications for this application. Typically sleep stages such as N1 are rare compared to N2 stages. When learning a predictive algorithm with very imbalanced classes, what classically happens is that the resulting system tends to never predict the rarest classes. One way to address this issue is to reweight the model loss function so that the cost of making an error on a rare sample is larger [16]. With an online training approach as used with neural networks, one way to achieve this is to employ balanced sampling, *i.e.* to feed the network with batches of data which contain as many data points from each class [35, 36, 12, 33]. This indeed prevents the predictive models to be biased towards the most frequent stages. Yet, such a strategy raises the question of the choice of the evaluation metric used. The standard *Accuracy* metric (Acc.) considers that any prediction mistake has the same cost. Imagine that N2 would represent 90 % of the data, predicting always N2 would lead to a 90 % accuracy, which is obviously bad. A natural way to better evaluate a model in the presence of imbalanced classes is to use the *Balanced Accuracy* (B. Acc.) metric. With this metric the cost of a mistake on a sample of type N2 is inversely proportional to the fraction of samples of type N2 in the data. By doing so, every sleep stage has the same impact on the final figure of merit [21].

Another statistical learning challenge regards the way transition rules are handled. Indeed, as the transition rules may impact the final decision of a scorer, a predictive model might take them into account in order to increase its performance. Doing so is possible by feeding the final classifier with the features from the neighboring time segments [33, 12, 36, 35]. This is referred to as *temporal sleep stage classification*.

Existing public sleep datasets contain PSG records with several EEG channels, and additional modalities such as EOG or EMG channels [26]. Although these modalities are used by human sleep experts in the sleep scoring process, seldom are they considered by automatic sleep scoring systems [21]. Focusing only on the EEG modality, it is natural to think that the multivariate nature of EEG data does carry precious information, yet it is often only used to cope with electrode removal or bad channels and not as a leverage to improve the algorithm. Indeed, the EEG community have designed a number of methods to increase the signal-to-noise ratio (SNR) of an effect of interest from a full array of sensors. Among these methods are so called linear spatial filters and include classical techniques such as PCA/ICA [27], Common Spatial Patterns for BCI applications [7] or beamforming methods for source localization [37]. Less classically and more recently various deep learning approaches have been proposed to learn from EEG data [24, 38, 40, 4] and some of these contributions use a first layer that boils down to a spatial filter [8, 9, 31, 23, 32, 22]. Yet using a deep network model to classify sleep stages on an array of sensors building on the idea of spatial filters has so far not been proposed in the literature.

This paper is organized as follows. First we introduce our end-to-end deep learning approach to perform temporal sleep stage classification using multivariate time series coming from multiple modalities (EEG, EOG, EMG). We furthermore detail how the temporal context of each segment of data can be exploited by our model. Then, we benchmark our approach on publicly available data and compare it to state-of-the-art sleep stage classification methods. To finish, we explore the dependencies of our approach regarding the spatial context, the temporal context and the amount of training data at hand.

Notation We denote by $X \in \mathbb{R}^{C \times T}$ a segment of multivariate time series, with its label $y \in \mathcal{Y}$. In the present application, X corresponds to a sample lasting 30 seconds and $\mathcal{Y} = \{W, N1, N2, N3, REM\}$. Here C refers to the number of channels and T to the number of time instants. $\{X_t, X_{t+1}, \cdots, X_{t+k-1}\}$ refers to an ordered sequence of k neighboring segments of signals. The classification task consists in learning a model

$\hat{f} : (\mathbb{R}^{C\times T})^{2k+1} \longrightarrow \mathcal{Y}$ that predicts $\hat{y}_t \in \mathcal{Y}$ given an input ordered sequence $\mathcal{S}_t^k = \{X_{t-k}, \cdots, X_t, \cdots, X_{t+k}\}$ of $2k+1$ neighboring segments of signal. If $k = 0$, the task boils down to the standard sleep stage classification problem. If $k > 0$, the task takes into account the temporal context, and we refer to it as *temporal sleep stage classification*.

2 Material and methods

In this section, we present a deep learning architecture to perform temporal sleep stage classification from multivariate and multimodal input time series. We first present the network architecture used to predict from multivariate time series without temporal context ($k = 0$). Then we describe the time distributed multivariate network proposed to perform temporal sleep stage classification ($k > 0$). Finally, we present and discuss the alternative state-of-the-art methods used for comparison in our experiments.

2.1 Multivariate Network Architecture

The deep network architecture we propose to perform sleep stage classification from multivariate time series without temporal context ($k = 0$) has three key features: linear spatial filtering to estimate so called *virtual channels*, convolutive layers to capture spectral features and separate pipelines for EEG/EOG and EMG respectively. This network constitutes a general feature extractor we denote by $Z : \mathbb{R}^{C\times T} \rightarrow \mathbb{R}^D$, where D is the size of the estimated feature space. Our network can handle various number of input channels and several modalities at the same time. The general architecture is represented in Fig. 1.

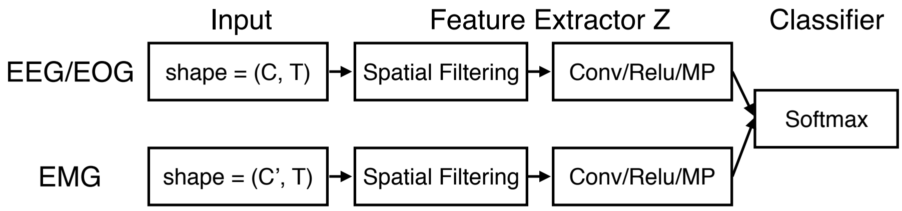

Figure 1: Network general architecture: the network processes C EEG/EOG channels and C' EMG channels through separate pipelines. For each modalitity, it performs spatial filtering and applies convolutions, non linear operations and max-pooling (MP) over the time axis. The outputs of the different pipelines are finally concatenated to feed a softmax classifier.

We now detail the different blocks of the network, which are summarized in Tab. 1. The first layer of the network is a time-independent linear operation that outputs a set of virtual channels, each obtained by linear combination of the original input channels. It implements a *spatial filtering* driven by the classification task to perform [8, 9, 31, 23, 32, 22].

In our experiments, the number of virtual channels was set to the number of input channels making the first layer a multiplication with a square matrix. This square matrix plays the same role as the unmixing matrix estimated by ICA algorithms. This step will be further discussed in the last section. Note that this first layer based on spatial filters can be implemented with a 2D valid convolution with kernels of shape $(C, 1)$, see layer 3 in Tab. 1.

Following this linear operation, the dimensions are permuted, see layer 4 in Tab. 1. Then two blocks of temporal convolution followed by non-linearity and max-pooling are consecutively applied. The parameters have been set for signals sampled at 128 Hz. In this case the number of time steps is $T = 128 \times 30 = 3840$. Each block first convolves its input signal with 8 estimated kernels of length 64 with stride 1 (~ 0.5 s of record) before applying a rectified linear unit, *a.k.a.* ReLU non-linearity $x \mapsto \max(x, 0)$ [25]. The outputs are then reduced along the time axis with a maxpooling layer (poolsize of 16, without overlap). The output of the two convolution blocks is finally passed through a dropout layer [30] which randomly prevents updates of 25% of the output neurons at each gradient step.

As represented in Fig. 1, we process jointly the EEG and EOG time series since these modalities are comparable in magnitudes and both measure similar signals, namelly electric potential up to a few hundreds of microvolts on the surface of the scalp. The same idea is used by EEG practitioners when the EOG channels are kept in the ICA decomposition to better reject EOG artifacts [19]. The EMG time series which have different statistical and spectral properties are processed in a parallel pipeline.

The resulting outputs are then concatenated to form the feature space of dimension D before being fed into a final layer with 5 neurons and a *softmax* non linearity to obtain a probability vector which sums to one. This final layer is referred to as a *softmax classifier* [14].

	Layer	Layer Type	# filters	# params	size	stride	Output dimension	Activation	Mode
	1	Input					(C, T)		
	2	Reshape					(C, T, 1)		
	3	Convolution 2D	C	C * C	(C, 1)	(1, 1)	(1, T, C)	Linear	
	4	Permute					(C, T, 1)		
Features	5	convolution 2D	8	8 * 64 + 8	(1, 64)	(1, 1)	(C, T, 8)	Relu	same
Extractor	6	maxpooling 2D			(1, 16)	(1, 16)	(C, T // 16, 8)		
	7	convolution 2D	8	8 * 8 * 64 + 8	(1, 64)	(1, 1)	(C, T // 16, 8)	Relu	same
	8	maxpooling 2D			(1, 16)	(1, 16)	(C, T // 256, 8)		
	9	Flatten					(C * (T // 256) * 8)		
	10	Dropout (50%)					(C * (T // 256) * 8)		
Classifier	11	Dense		5 * (C * T // 64 * 8)			5	Softmax	

Table 1: Detailed architecture for the feature extractor for C EEG channels with time series of length T. The same architecture is employed for C' EMG channels. When both EEG / EOG and EMG are considered, the outputs of the dropout layers are concatenated and fed into the final classifier. The number of parameters of the final dense layer becomes thus equal to $5 \times ((C + C') \times (T/256) \times 8)$.

2.2 Time Distributed Multivariate Network

In this paragraph, we describe the *Time Distributed Multivariate Network* we propose to perform *temporal sleep stage classification* ($k > 0$). It builds on the *Multivariate Network Architecture* presented previously and distributes it in time to take into account the temporal context. Indeed a sample of class N2 is very likely to be close to another N2 sample, but also to an N1 or an N3 sample [18].

To take into account the statistical properties of the signals before and after the sample of interest, we propose to aggregate the different features extracted by Z on a number of time segments preceding or following the sample of interest. More formally, let $\mathcal{S}_t^k = \{X_{t-k}, \cdots, X_t, \cdots, X_{t+k}\}$ be a sequence of $2k+1$ neighboring samples (k samples in the past and k samples in the future). Distributing in time the features extractor consists in applying Z to each sample in $\mathcal{S}_t^k$ and aggregating the $2k+1$ outputs forming a vector of size $D(2k+1)$. Then, the obtained vector is fed into the final softmax classifier. This is summarized in Fig. 2.

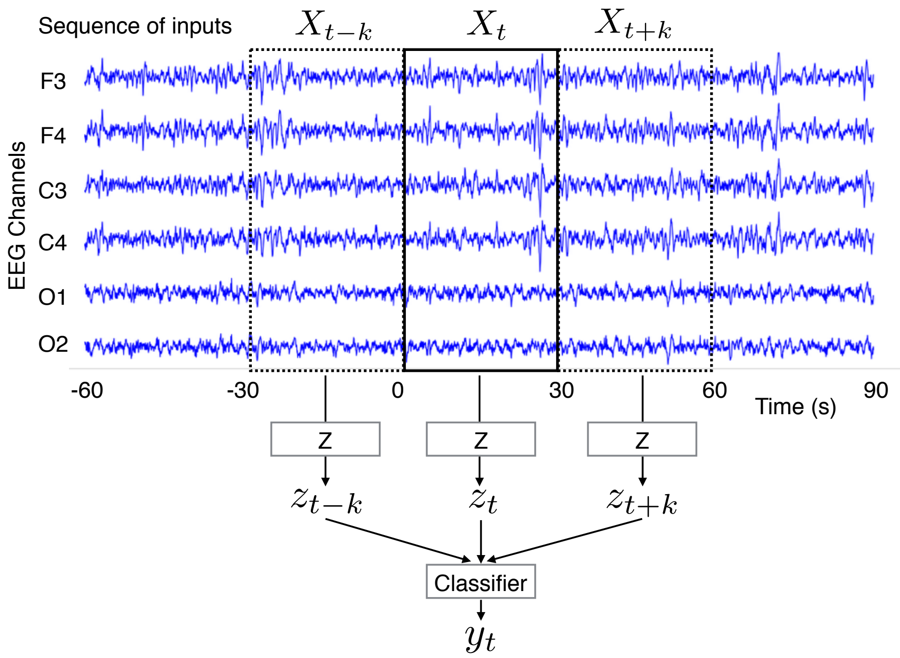

Figure 2: Time distributed architecture to process a sequence of inputs $\mathcal{S}_t^k = \{X_{t-k}, \cdots, X_t, \cdots, X_{t+k}\}$ with $k = 1$. X_k stands for the multivariate input data over 30 s that is fed into the feature extractor Z. Features are extracted from consecutive 30 s samples: $X_{t-k}, \cdots, X_t, \cdots, X_{t+k}$. Then the obtained features are aggregated $[z_{t-k}, \cdots, z_t, \cdots, z_{t+k}]$. The resulting aggregation of features is finally fed into a classifier to predict the label y_t associated to the sample to classify X_t

2.3 Training

Training was performed by minimizing the categorical crossentropy with a stochastic gradient descent using minibatches of data. Yet, to be able to learn to discriminate under-represented classes (typically W and N1 stages) and since we are interested in optimizing the balanced accuracy, we propose to balance the distribution of each class in minibatches of size 128. As we have 5 classes it means that during training, each batch has about 20% of samples of each class. The *Adam* optimizer [20] is used for optimization with the following parameters $\alpha = 0.001$ (learning rate), $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

An early stopping callback on the validation loss with patience of 5 epochs was used to stop the training process when no improvements were detected. Weights were initialized with a normal distribution with mean $\mu = 0$, and standard deviation $\sigma = 0.1$. Those values were obtained empirically by monitoring the loss during training. The implementation was written in *Keras* [11] with a *Theano* backend [34].

The training of the time distributed network was done in two steps. First, we trained the multivariate network, especially its feature extractor part Z_t without temporal context ($k = 0$). The trained model was then used to set the weights of the feature extractor distributed in time. Second, we freezed the weights of the feature extractor distributed in time and we trained the final softmax classifier with aggregated features.

2.4 Related work and compared approaches

We now introduce the two state-of-the-art approaches that we used for comparison with our approach: a gradient boosting classifier [13] trained on hand-crafted features and a convolutional network trained on univariate time series following the approach of [36].

2.4.1 Features based approach

The *Gradient Boosting* model was learnt on hand-crafted features: time domain features and frequency domain features computed for each input sensor as described in [21]. More precisely, we extracted from each channel power and relative power in 5 bands: δ $(0.5-4.5\ \text{Hz})$, θ $(4.5-8.5\ \text{Hz})$, α $(8.5-11.5\,\text{Hz})$, σ $(11.5-15.5\,\text{Hz})$, β $(15.5-30\,\text{Hz})$, giving both 5 features. We furthermore extracted power ratios between these bands (which amount for $5 \times 4/2 = 10$ supplementary features) and spectral entropy features as well as statistics such as mean, variance, skewness, kurtosis, 75% quantile. This gives in the end a set of 26 features per channel.

The implementation used is from the *XGBoost* package [10], which internally employs decisions trees. This model is known for its high predictive performance, robustness to outliers, robustness to unbalanced classes and parallel search of the best split. Training was performed by minimizing also the categorical crossentropy. The maximum number of trees in the model was set to 1000. An early stopping callback on the validation categorical crossentropy with patience equal to 10 was used to stop the training when no improvement was observed. Training never led to 1000 trees in a model.

The model has several hyper-parameters that need to be tuned to improve classification performances and cope with unbalanced classes. To find the best hyper-parameters for each experiment, we performed random searches with the *hyperopt* Python package [5]. Concretely, we considered only the data from the training and validation subjects at hand. For each set of hyper-parameters, we trained and evaluated the classifier on data from 5 different splits of training and evaluation subjects (80% for training 20% for evaluation). The search was done with 50 sets of hyper-parameters and the set which achieved the best balanced accuracy averaged on the 5 splits was selected.

The following parameters were tuned: learning rate in interval $[0.01, 0.5]$, the minimum weight of a child tree in set $\{1, 2, \cdots, 10\}$, the maximum depth of trees in $\{1, 2, \cdots, 10\}$, the regularization parameter in $[0, 1]$, the subsampling parameter in $[0.5, 1]$, the sampling level of columns by tree in $[0.5, 1]$.

2.4.2 Convolutional network for univariate time series

The approach by *Tsinalis et al. 2016* [36] was reimplemented according to the paper details and used for comparison with a deep learning classifier. It consists in a deep convolution network that processes univariate time series (a single EEG signal). The approach originally takes into account the temporal context, by feeding the network with 150 s of signals, *i.e.* the sample to classify plus the 2 previous and 2 following samples. When used without temporal context in the experiments, the network is fed with 30 s samples.

Training was performed by minimizing the categorical crossentropy, and a similar balanced sampling strategy with *Adam* optimizer was used. An additional ℓ_2 regularization set to 0.01 was applied onto the convolution filters [36]. The code was written in *Keras* [11] with a *Theano* backend [34].

3 Experiments

In this section, we first introduce the dataset and the preprocessing steps used. Then we present the experiments which aim at (i) establishing a general benchmark under a common spatial and temporal context and stressing the importance of considering the multivariate structure of the data, (ii) studying the influence of the type of channels used, (iii) evaluating the gain obtained by using the temporal context and (iv) evaluating the impact of the quantity of training data.

3.1 Data, preprocessing, performance metrics

Data used in our experiments is the publicly available MASS dataset - session 3 [26]. It corresponds to 62 night records, each one coming from a different subject. Because of preprocessing issues we removed the record *01-03-0034*. Each record contains data from 20 EEG channels which were referenced with respect to the $A2$ electrode. We did not modify the referencing scheme, hence removed the $A2$ electrode from our study. Each record also includes signals from 2 EOG and 3 EMG channels that we considered as additional modalities.

Every time series was downsampled to a sampling rate of 128 Hz. The signals from EEG sensors were low-pass filtered with a 30 Hz cut-off frequency in order for the deep network to have access to the same frequency content as the feature based approach. The data extraction and the filtering steps were performed with the *MNE software* [15]. The filter employed was a zero-phase finite impulse response (FIR) filter with transition bandwidth of approximately 7 Hz. Sleep stages were marked according to the AASM rules by a single sleep expert per record [26, 18]. We additionnally did not consider the 10 first and 10 last 30 s samples of each record for evaluating the models. This enables to compare models which exploit the temporal context with models which do not.

The time series fed into the different neural networks were additionnaly standardized. Indeed, for each channel, every 30 s sample is standardized individually such that it has zero mean and unit variance. For the specific task of sleep stage classification this is particularly relevant since records are carried out over nearly 8 hours. During such a long period the recording conditions vary such as skin humidity, body temperature, body movements or even worse electrode contact loss. Giving to each 30 s time series the same first and second order moments enables to cope with this likely covariate shift that may occur during a night record. This operation only rescales the frequency powers in every frequency band, without altering their relative amplitudes where the discriminant information for the considered sleep stage classification task lies (see Parseval's theorem). Note that this preprocessing step can be done online before feeding the network with a batch of data.

Unless stated otherwise, data from 41 night records were used for training, 10 for validation and 10 for testing (performance evaluation). Each experiment was carried out 5 times shuffling the records for training, validation and testing to reduce variance in metric evaluation.

3.2 Experiment 1: The importance of thinking "multivariate"

In this experiment, we perform a general benchmark of our approach against *Tsinalis et al. 2016* and *Gradient Boosting* under a common spatial and temporal context.

Only time series coming from the channels Fz and Cz are considered here. Two versions of the predictive models are used: *Univariate* and *Multivariate*. First, the three predictive models were fed with the time series or the features from the derivation Fz-Cz that was computed manually. This version is referred to as *Univariate*. Second, *Gradient Boosting* and our approach were fed with the time series or features from the derivations Fz-A2 and Cz-A2, *i.e.*, the original time series of the dataset with pre-computed references. This version is referred to as *Multivariate*. We recall that *Tsinalis et al. 2016* network was originally fed with 150 s samples. Yet in this benchmark, we fed it with 30 s samples to give every method access to the same temporal context and thus the same amount of information.

Finally, concerning the proposed approach, the experiment was carried out using either balanced sampling as done in *Tsinalis et al. 2016* or random sampling as for *Gradient Boosting*. The *Balanced Accuracy* (*B. Acc*), *Accuracy* (*Acc.*) and *Confusions Matrices* (*C.M.*) were used as performance metrics for each predictive model. Regarding the *C.M.*, they were obtained by (i) normalizing the *C.M.* evaluated per testing subject such that its rows sum up to 1, (ii) computing the average *C.M.* over all testing subjects. The results are reported in Fig. 3.

What can be observed is that the *Multivariate* models exhibit higher performances than the *Univariate* models in terms of both *B. Acc.* and *Acc.*. Indeed, the confusion matrices indicate that all sleep stages are better indentified when using the *Multivariate* version of the network proposed here.

The predictive models trained with balanced sampling (resp. random sampling) exhibit a higher *B. Acc.* (resp. *Acc.*). Indeed, concerning the confusion matrices, the predictive models trained with balanced sampling identify reasonably well all stages, including $N1$, whereas the predictive models trained with random sampling identify very well (even slightly better) the most common stages (W, $N2$, REM), yet exhibit both a decrease in their ability to predict $N3$ correctly and a strong decrease in their ability to predict $N1$ correctly.

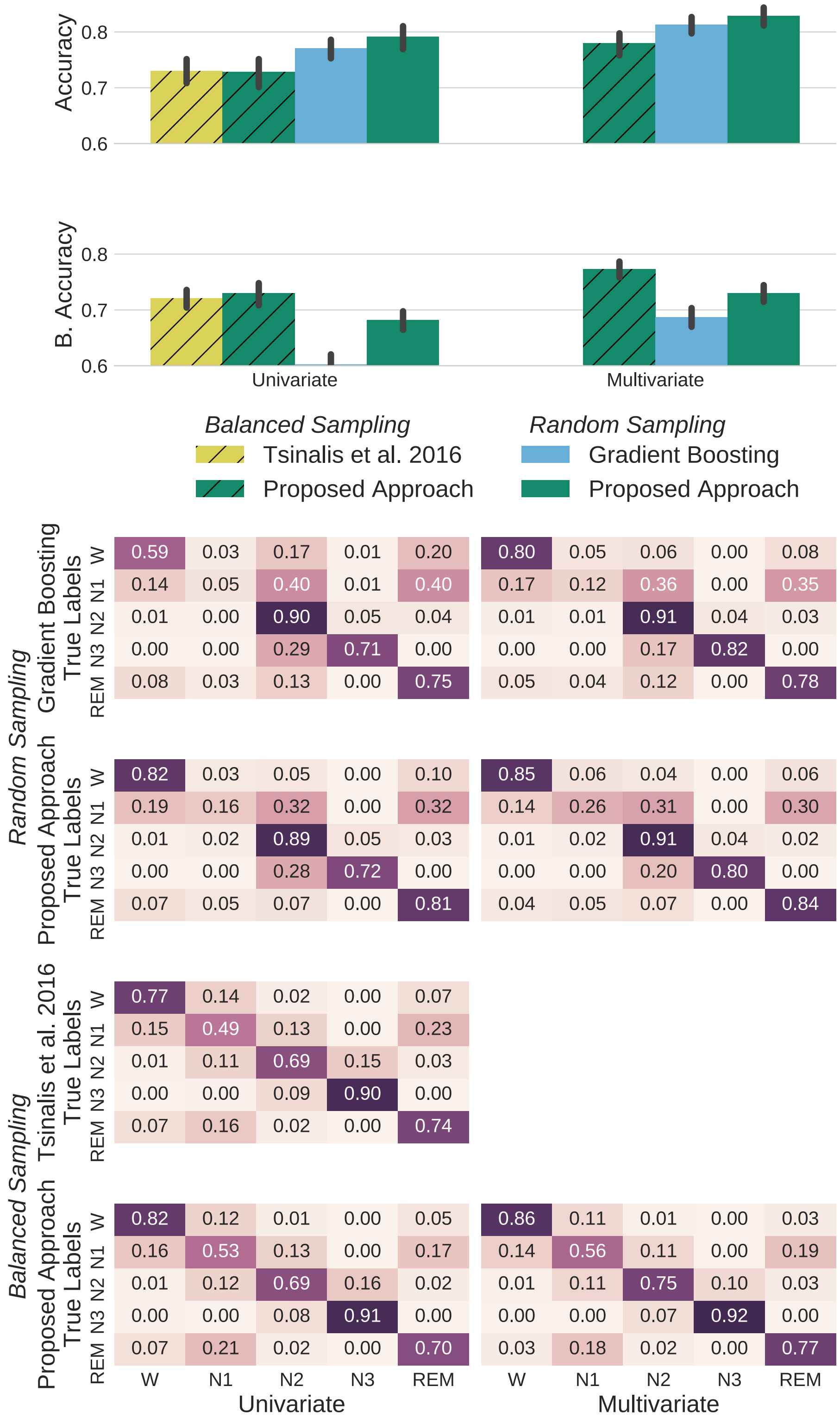

Figure 3: General benchmark on Fz-Cz derivation: considering the *Multivariate* character of the data induces a boost in classification performances. From top to bottom: Acc, B. Acc., C.M. for *Gradient Boosting* and our approach trained with random sampling, *Tsinalis et al 2016* and our approach trained with balanced sampling. Left column: *Univariate* version of predictive models: the input of classifiers is univariate *i.e.* they are fed with features or time series from the Fz-Cz derivation. Right column: *Multivariate* version of predictive models: the input of classifiers are multivariate, they are fed with features or time series from the Fz-A2 and Cz-A2 derivations.

3.3 Experiment 2: More sensors increase performance

In this experiment, we investigated the influence of the multivariate spatial context on the performance of our approach.

We considered 7 different configurations of EEG sensors which varied both in the number of recording sensors from 2 to 20 as well as in their positions over the head. We report the classification results for each configuration in Fig. 4.

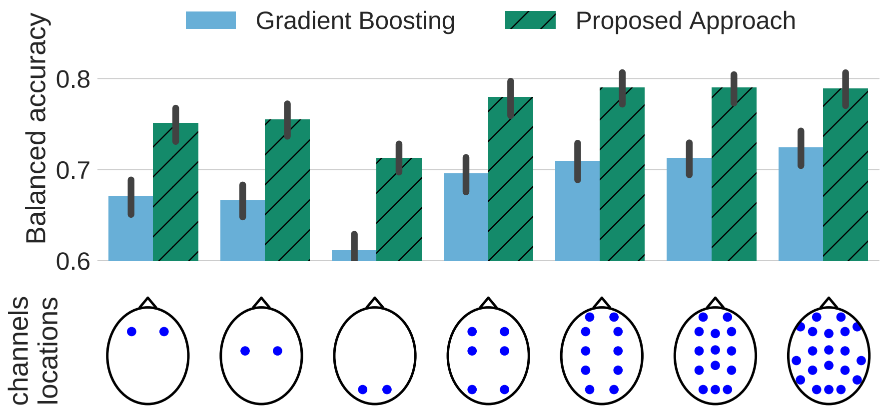

Figure 4: Influence of channel selection on the classification performances: increasing the number of EEG sensors increases B. Acc.

One observes that both *Gradient Boosting* and our approach benefit from the increased number of EEG sensors. However, the *B. Acc.* obtained with our approach does not improve once we have 6 well distributed channels. This is certainly due to the redundancy of the EEG channels, yet more channels could make on some data the model more robust to the presence of bad sensors. First, this demonstrates that it is worth adding more EEG sensors, but up to a certain point. Second, it shows that our approach exploits well the multivariate nature of signals to improve classification performances. Third, it shows that the channel agnostic features extractor, *i.e.* the use of the spatial projection and the features extractor is a good option to fully exploit the spatial distribution of the sensors.

Restricting the number of EEG channels to 6 and 20, we further investigated the influence of additionnal modalities (EOG, EMG). Classification results are provided in Fig. 5.

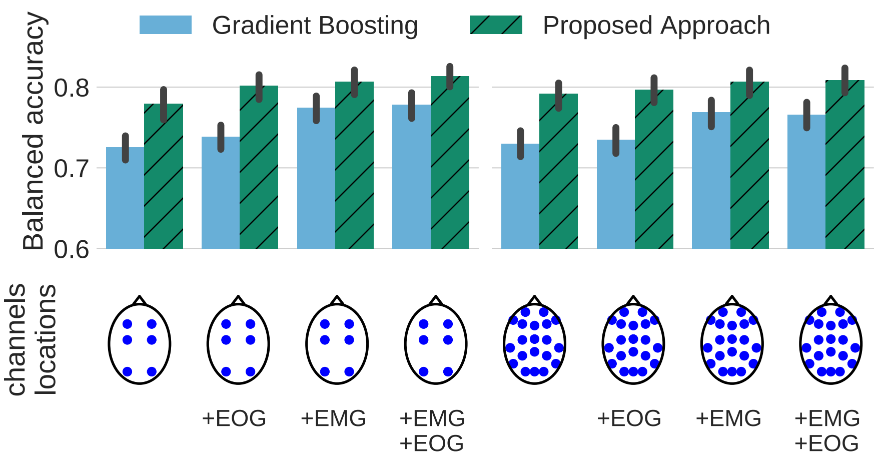

Figure 5: Influence of additional modalities on the classification performances: adding EOG and EMG induces a boost in performance

Considering additional modalities also increases the classification performances of the considered classifiers. It gives them a significative boost of performance, especially when the EMG modality is considered. This means that both approaches successfully integrate the new features with the previous ones.

This suggests that our feature extractor was sufficiently data agnostic and versatile to handle both modalities. Finally, it again stresses the importance of considering the spatial context, here the additionnal modalities, to improve classification performances.

Interestingly, the boost of performance is more important in the 6 channel setting rather than in the 20 channel setting. We further observe that both EEG configuration with EOG and EMG modalities reach the same performances. Thus, the use of additional modalities compensate the use of a larger spatial context in this situation. Practically speaking, to obtain the highest performances at a reduced computational cost, one shall consider few well located EEG sensors with additional modalities.

3.4 Experiment 3: Temporal context boosts performance

In this experiment, we investigate the influence of the temporal context on the classification performances and demonstrate that considering the data from the neighboring samples increases classification performances especially if the spatial context is limited.

To demonstrate this, we considered the spatial configurations with 2 frontal EEG channels, 6 EEG channels, and 6 EEG channels plus 2 EOG and 3 EMG channels. We varied the size of the temporal input sequence S_k from $k = 0$, *i.e.* without temporal context, up to $k = 5$, *i.e.* with a temporal context of 150 s preceding and following each sample to classify. The classification results are reported in Fig. 6.

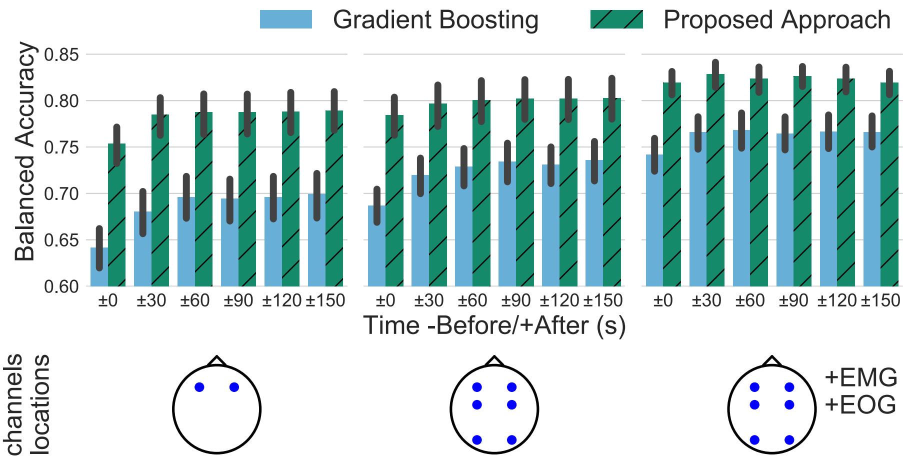

Figure 6: Influence of temporal context: considering the close temporal context induces a boost in performance especially when the spatial context is limited.

We observe in Fig. 6 that considering the close temporal context induces a boost in classification performances. The gain strongly depends on the spatial context taken into account. Indeed, our model trained on 2 frontal channels with $-30/+30$ s of context achieves similar performances than with the 6 EEG channel montage without temporal context. On the other hand, when considering an extended spatial context, the gain due to the temporal context turns out to be limited, as the performances of our approach or *Gradient Boosting* with the 6 EEG channels + 2 EOG and 3 EMG channels suggest.

This experiment shows that our model clearly benefits from the temporal context, which validates the use of the features extractor distributed in time. It furthermore offers some insights about the influence of the close temporal context in relation to the spatial context.

3.5 Experiment 4: More training data boost performance

In this experiment, we investigated the influence of the quantity of data on the classification performances of our approach.

To do this we considered the spatial configurations with 2 frontal EEG channels, 6 EEG channels, and 6 EEG channels plus 2 EOG and 3 EMG channels. Concretely, we varied the number of training records n in $\{3, 12, 22, 31, 41\}$. We considered the same number of records for validation and testing as previously, *i.e.* 10. We furthermore carried out the experiments over 5 random splits of training, validation and testing subjects. The classification results are reported in Fig. 7.

Every algorithm with any spatial context exhibits an increase in performance when there is more training data. *Gradient Boosting* is more resilient to the little data situation, and its performances for each spatial context increase only by 0.05 points with more data. On the other hand, our deep learning model exhibits another behavior: the increase in performances is greater than 0.1 for any spatial context, when the number of training records goes from 3 to 41.

Furthermore, it appears that having few training records but an extended spatial context delivers as good performances as with many training records and few channels. Said differently, a rich spatial context can compensate for the scarcity of training data. Indeed, the input configuration with 6 EEG channels plus 2 EOG

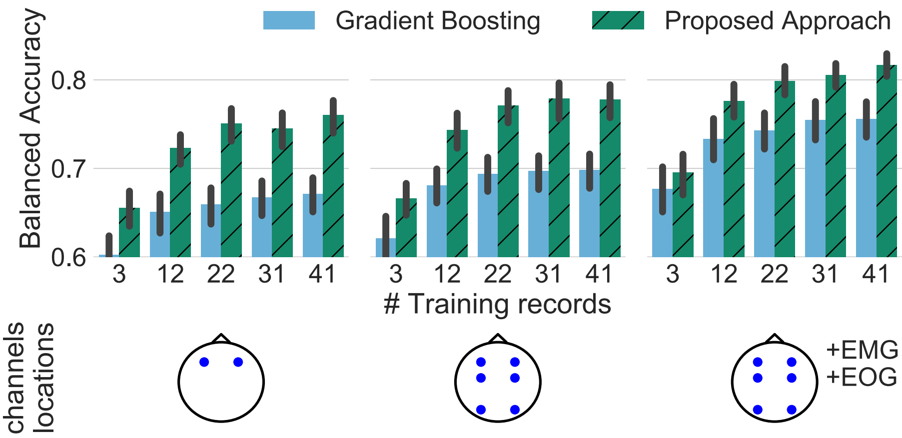

Figure 7: Influence of the number of training records: the more training records the better performances are.

and 3 EMG channels with only 12 training subjects (right sub-figure) reaches the same performance as the 2 EEG channels input configuration (left sub-figure) with 41 training subjects.

3.6 Experiment 5: Opening the model box

In this experiment, we investigated the sensitivity of our models to the different frequency bands of interest for sleep stage classification. To do so, we trained a model in a standard configuration but made it predict on data filtered in different frequency bands. Such an operation, referred to as occlusion sensitivity has been successfully used to better understand how deep neural networks classify images [39]. This way, not only do we show the dependency of our models towards the bands considered but also we do recover information about the association of sleep stages to specific frequency contents.

We used the deep neural network that takes as input EEG data coming from 6 channels uniformly located over the head (2 frontal, 2 central and 2 occipital channels). We made the network predict on usual data and on filtered data using the following frequency bands: δ ($0.5 - 4.5$ Hz), θ ($4.5 - 8.5$ Hz), α ($8.5 - 11.5$ Hz), σ ($11.5 - 15.5$ Hz), β ($15.5 - 30$ Hz). The operation was repeated 5 times randomizing the training, evaluation and testing records. The confusion matrices associated to the different filterings are reported in Fig. 8.

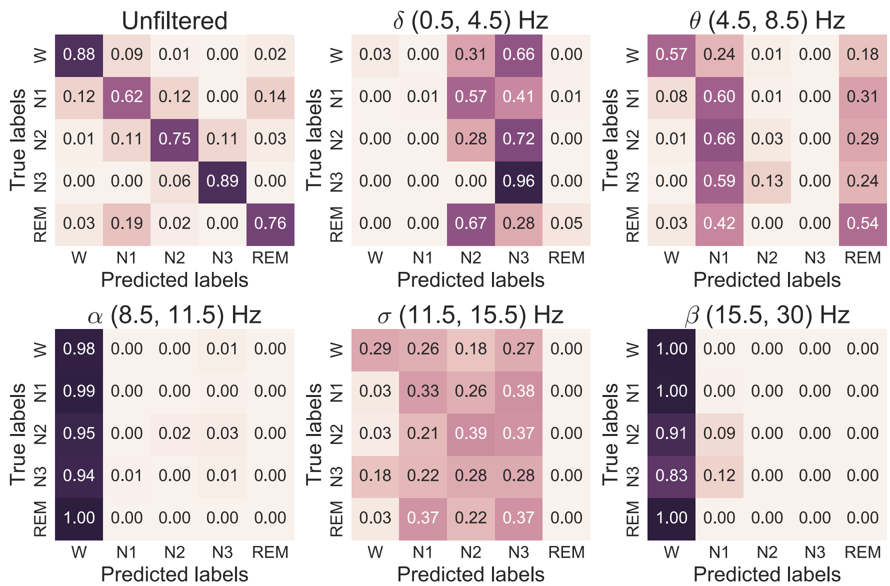

Figure 8: Prediction on filtered data: confusion matrices associated to unfiltered and filtered signals from testing records.

Filtering the testing records alters the confusion matrices in such a way that only specific stages are mostly predicted by the network compared to unfiltered data. For example, when filtering the data in the δ frequency band, the network predicted almost only N2 and N3 stages. This confirms that N2 and N3 stages are dominated by low frequency content taking the form of slow oscillations or K-complex. When filtering within the α

frequency band, the network only predicted the W stage, traditionaly associated to α waves. Finally, when filtering in the σ frequency band, the network often predicted N2 stages, where spindles (with frequencies in $10-15$ Hz) are observed. Despite the black-box nature of this predictive approach, this procedure allows to open the box and to reveal interesting insights about the data.

4 Discussion

We have here detailed and evaluated a network architecture dedicated to temporal sleep stage classification.

Our architecture was designed to handle multivariate inputs. Motivated by simplicity, we chose the number of virtual channels equal to the number of input channels. Yet, this constitutes a degree of freedom one may play with to increase the performances of the network as was explored in [9]. We then decided to consider a simple and versatile 2 layer architecture for this study. Considering fewer or more layers was explored but did not deliver any extra gain in performance. We furthermore opted to perform spatial and temporal convolutions strictly separately. By doing so we remplaced possible 2D expensive convolutions by a 1D spatial convolution and a 1D temporal convolution. Such low rank spatio-temporal convolution strategy turned out to be successful in our experiments.

Regarding the dimensions of the convolution filters and pooling regions, our approach was motivated by the ability of neural networks to learn a hierarchical representation of input data, extracting low level and small scale patterns in the shallow layers and more complex and large scale patterns in the deep layers. Our strategy is quite different from [36, 33] which use large temporal convolution filters. Despite the use of shorter filters, Fig. 3 and Fig. 8 demonstrate that our architecture is able to discriminate stages with low frequency content, such as $N3$, from stages with higher frequency content such as $N2$ due to the presence of spindles, or even from W and $N1$ with the presence of α ($8-12$Hz) bursts. Besides, our proposed architecture turns out to be data agnostic and handles well both EEG, EOG and EMG signals as shown by the results of experiment 2, see Fig. 4 and Fig 5.

The complexity of our network and its number of parameters are quite small thanks to specific architecture choices. The overal network does not exhibit more than $\sim 10^4$ parameters when considering an extended spatial context, and not more than $\sim 10^5$ parameters when considering both an extended spatial context and an extended temporal context. This is quite simple and compact compared to the recent approaches in [36] which has up to $\sim 14.10^7$ parameters and [33] which exhibits $\sim 6.10^5$ parameters for the feature extractor and 2.10^7 parameters for the sequence learning part using BiLSTM. This significant difference with [36] is mainly due to our choice of using small convolution filters (64 time steps after low pass filtering and downsampling), large pooling regions (pooling over 16 time steps) according to the 128 Hz sampling frequency and removing the penultimate fully connected layers before the final softmax classifier. Such a strategy has already been successul in computer vision [29] and EEG [22]. Regarding the comparison with [33], our approach is different in the way that it builds a features representation as explained previously but also in the way how it processes the temporal context without using any recurrent neural network.

Our architecture choice to grasp the temporal context is straightforward as it only relies on the aggregation of temporal features and a softmax classifier. Such a choice, enabled us to measure the influence of the close temporal context. Yet, recent approaches have proposed more complex strategies to integrate the temporal context with LSTM unit cells or Bi-LSTM unit cells [12, 33, 17]. We believe that our approach is versatile enough to be coupled with a more complex time context integration process. Doing so might deliver significative gains in performances also with an extended spatial context, but shall be addressed in future work.

Fig. 7 raises an important question: how much data is needed to establish a correct benchmark of predictive models for sleep stage classification. This is particularly interesting concerning the deep learning approaches. Indeed, the *Gradient Boosting* handles quite well the small data situation and does not exhibit a huge increase in performances with the increase of the number of training records. On the contrary our approach delivers particularly good performances if enough training data are at hand. Extrapolation of the learning curves (performance as a function of the number of training records) in Fig. 7 suggests that one could expect better performances if more data were accessible. This forces us to reconsider the way we compare predictive models when training dataset sizes differ between experiments since the quantity of training data plays the role of a hyper-parameter for some algorithms like ours. Some algorithms become indeed better when more data are available (see for example Fig. 1 in [3]).

Our first experiment raised a particularly important point concerning the metrics used to measure the performances of sleep stage classifiers. As we observed the choice of sampling strategies employed during online learning impacts the evaluation metrics and conversely the choice of metrics should motivate the choice of sampling strategies. Indeed, balanced sampling should be used to optimize the balanced accuracy of the model. On the other hand, random sampling should be used to boost the accuracy. All our experiments were driven by the objective to maximize the balanced accuracy of the deep network, *i.e.* by the objective to correctly identify *all* sleep stages. Nonetheless, for a specific clinical application, one may decide that errors on a minor stage, such as $N1$, are not so dramatic and hence prefer to train the network with random batches of data.

5 Conclusion

In this study we introduced a deep neural network to perform temporal sleep stage classification from multimodal and multivariate time series. The model pools information from different sensors thanks to a linear spatial filtering operation and builds a hierarchical features representation of PSG data thanks to temporal convolutions. It additionally pools information from different modalities processed with separate pipelines. The network trained with balanced sampling reaches top performances quantified by balanced accuracy. It benefits from increasing the number of EEG channels taken into account and from considering the EMG channels as additional modalities. Furthermore, the proposed approach performances are boosted when some temporal context is taken into account. The effect of temporal context is significantly stronger when considering a limited spatial context rather than an extended one. Finally, we demonstrated that the proposed approach delivers state-of-the-art performances provided it is trained on a sufficiently large number of samples.

Acknowledgement

This work was supported in part by the french Association Nationale de la Recherche et de la Technologie (ANRT) under Grant 2015 / 1005.

References

[1] Aboalayon, K., Faezipour, M., Almuhammadi, W., and Moslehpour, S. Sleep Stage Classification Using EEG Signal Analysis: A Comprehensive Survey and New Investigation. *Entropy 18*, 9 (2016), 272.

[2] Allan Hobson, J. A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. *Electroencephalography and Clinical Neurophysiology 26*, 6 (June 1969), 644.

[3] Banko, M., and Brill, E. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* (Stroudsburg, PA, USA, 2001), ACL '01, Association for Computational Linguistics, pp. 26–33.

[4] Bashivan, P., Rish, I., Yeasin, M., and Codella, N. Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks. *ICLR* (2016), 1–15.

[5] Bergstra, J., D., Y., and D., C. D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *ICML* (2013).

[6] Berthomier, C., Drouot, X., Herman-Stoïca, M., Berthomier, P., Prado, J., Bokar-Thire, D., Benoit, O., Mattout, J., and D'Ortho, M.-P. Automatic analysis of single-channel sleep EEG: validation in healthy individuals. *Sleep 30*, 11 (2007), 1587–1595.

[7] Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., and Muller, K. R. Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine 25*, 1 (2008), 41–56.

[8] Cecotti, H., and Gräser, A. Convolutional Neural Network with embedded Fourier Transform for EEG Classification. In *19th International Conference on Pattern Recognition* (Dec 2008), pp. 1–4.

[9] Cecotti, H., and Gräser, A. Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 3 (Mar. 2011), 433–445.

[10] Chen, T., and Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining* (2016), ACM, pp. 785–794.

[11] Chollet, F. Keras. `https://github.com/fchollet/keras`, 2015.

[12] Dong, H., Supratak, A., Pan, W., Wu, C., Matthews, P. M., and Guo, Y. Mixed Neural Network Approach for Temporal Sleep Stage Classification. *arXiv:1610.06421v1 1* (2016).

[13] Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics 29* (2000), 1189–1232.

[14] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[15] Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Parkkonen, L., and Hämäläinen, M. S. MNE software for processing MEG and EEG data. *NeuroImage 86* (2014), 446–460.

[16] He, H., and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering 21*, 9 (Sept 2009), 1263–1284.

[17] Hochreiter, S., and Schmidhuber, J. Long Short-Term Memory. *Neural Comput. 9*, 8 (Nov. 1997), 1735–1780.

[18] Iber, C., Ancoli-Israel, S., Chesson, A., and Quan, S. F. The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specification, 2007.

[19] Joyce, C. A., Gorodnitsky, I. F., and Kutas, M. Automatic Removal of Eye Movement and Blink Artifacts from EEG Data using Blind Component Separation. *Psychophysiology 41*, 2 (2004), 313–325.

[20] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *CoRR abs/1412.6980* (2014).

[21] Lajnef, T., Chaibi, S., Ruby, P., Aguera, P.-E., Eichenlaub, J.-B., Samet, M., Kachouri, A., and Jerbi, K. Learning Machines and Sleeping Brains: Automatic Sleep Stage Classification using Decision-Tree Multi-Class Support Vector Machines. *Journal of Neuroscience Methods 250*, November (2015), 94–105.

[22] Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., and Lance, B. J. EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces. *arXiv:1611.08024v2* (2016), 1–20.

[23] Manor, R., and Geva, A. B. Convolutional Neural Network for Multi-Category Rapid Serial Visual Presentation BCI. *Frontiers in Computational Neuroscience 9*, December (2015), 146.

[24] Mirowski, P., Madhavan, D., LeCun, Y., and Kuzniecky, R. Classification of Patterns of EEG Synchronization for Seizure Prediction. *Clinical Neurophysiology 120*, 11 (2009), 1927–1940.

[25] Nair, V., and Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML* (2010), pp. 807–814.

[26] O'Reilly, C., Gosselin, N., Carrier, J., and Nielsen, T. Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research. *Journal of Sleep Research 23*, 6 (2014), 628–635.

[27] Parra, L. C., Spence, C. D., Gerson, A. D., and Sajda, P. Recipes for the Linear Analysis of EEG. *NeuroImage 28*, 2 (2005), 326 – 341.

[28] Rosenberg, R. S., and Van Hout, S. the American Academy of sleep Medicine Inter-scorer Reliability Program: Sleep Stage Scoring. *Journal of Clinical Sleep Medicine 10*, 4 (2014), 447–454.

[29] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for Simplicity: The All Convolutional Net. *ICLR* (2015), 1–14.

[30] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *JMLR 15* (2014), 1929–1958.

[31] Stober, S., Cameron, D. J., and Grahn, J. a. Using Convolutional Neural Networks to Recognize Rhythm Stimuli from Electroencephalography Recordings. *Advances in Neural Information Processing Systems 27* (2014), 1449–1457.

[32] Stober, S., Sternin, A., Owen, A. M., and Grahn, J. A. Deep Feature Learning for EEG Recordings. *arXiv:1511.04306v4* (2016), 1–24.

[33] Supratak, A., Dong, H., Wu, C., and Guo, Y. DeepSleepNet: a Model for Automatic Sleep Stage Scoring based on Raw Single-Channel EEG. *arXiv:1703.04046* (2017), 1–10.

[34] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints abs/1605.02688* (May 2016).

[35] Tsinalis, O., Matthews, P. M., and Guo, Y. Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders. *Annals of Biomedical Engineering 44*, 5 (2016), 1587–1597.

[36] Tsinalis, O., Matthews, P. M., Guo, Y., and Zafeiriou, S. Automatic Sleep Stage Scoring with Single-Channel EEG Using Convolutional Neural Networks. *arXiv:1610.01683* (2016), 1–10.

[37] Veen, B. D. V., Drongelen, W. V., Yuchtman, M., and Suzuki, A. Localization of Brain Electrical Activity via Linearly Constrained Minimum Variance Spatial Filtering. *IEEE Transactions on Biomedical Engineering 44*, 9 (Sept 1997), 867–880.

[38] Wulsin, D. F., Gupta, J. R., Mani, R., Blanco, J. A., and Litt, B. Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement. *Journal of Neural Engineering 8*, 3 (2011), 036015.

[39] Zeiler, M. D., and Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I* (2014), pp. 818–833.

[40] Zheng W.L. , Zhu J.Y., Peng Y., Lu B.L. EEG Based Emotion Classification using Deep Belief Networks. In *ICASSP* (2014).