Detecting Sleep Apnea from Raw Physiological Signals

Sparse Representations Challenge

Clément Grisi
Master MVA
École des Ponts Paristech
grisi.clement@gmail.com

Louis Bouvier
Master MVA
École des Ponts Paristech
louis.bouvier@eleves.enpc.fr

Abstract

*In this report, we emphasize our work for the ENS data challenge provided by Dreem which aims at detecting sleep apnea events from polysomnography (PSG) signals. Through iterations in model definition and data representation, we highlight the influence of a particular focus on some specificities of the signals, as well as on the data imbalance and on the post-processing to get a **0.6503** score in the private academic leader-board (rank 1) in the limited time of the MVA course. Our code for the part on deep learning models 5 is publicly available at* https://github.com/clementgr/detect-sleep-apnea.

1. Introduction

Sleep apnea is a widespread disorder affecting about 25 percent of male and 10 percent of female. Medically speaking, it is diagnosed when the Apnea-Hypopnea Index (AHI) of a patient is greater than 5, which means that more than 5 apnea (cessation of breathing) or hypopnea (difficulties of breathing) events occur within one hour. Both of them last more than 10 seconds by definition. One important detail to notice is that those disorders can be central or obstructive. In the case they are obstructive, the patient still has abdominal contractions, but the air flow as well as the saturation in O_2 are altered [1]. This detail is important to choose which signal we may start working with before building a model on the whole PSG.

Now that we are aware of the topic of this competition, we understand to which extent the automatic detection of apnea-hypopnea would be a boon on diagnosis (saving time, resources, and fostering alternatives to sometimes non-consensual annotations). Besides, we are able to build our models and assumptions on some key aspects inherent in the medical definitions. Indeed, for instance, we have in mind the diverse and complex symptoms of spleep apnea, as well as its minimum duration, both having a direct impact on our algorithms and data processing.

Another prior source of knowledge we introduce before considering the data provided is a recent review of machine learning attempts to address the sleep apnea detection [2]. Thanks to this paper, we have an overview of the features usually selected to represent PSG signals, their pre-processing, the common algorithms suggested for this task, a comparison of their results and a discussion of the assets and liabilities of the approaches published between 2008 and 2018. The two main architectures reaching state of the art performance are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), sometimes hybridised. Before experimenting with those latter, we start with tests on simpler models, extracting hand-crafted features from the raw data. We successively highlight the limits of those simpler approaches, paving the way for deep learning architectures. We eventually discuss their strengths and limits and suggest some future work.

2. Problem Definition

2.1. Dataset

The dataset used is provided by Dreem. It consists of 4400 training and 4400 testing samples from 44 nights recorded with a polysomnography and scored for apnea events by a consensus of human experts. For each of the 44 nights, 200 windows (without intersection) were sampled. Each of these windows contains 90 seconds of signal from the following 8 physiological signals sampled at 100Hz:

- Abdominal belt: abdominal contraction
- Airflow: respiratory airflow from the subject
- PPG (Photoplethysmogram): cardiac activity
- Thoracic belt: record thoracic contraction
- Snoring indicator
- SPO$_2$: O2 saturation of the blood
- C4-A1: EEG derivation
- O2-A1: EEG derivation

The associated segmentation masks are sampled at 1Hz and contain 90 labels (0 = no event, 1 = apnea event). In order to be able to compare the different deep learning models we develop, we use the same train/val split across all deep learning related experiments: we set aside 30% of training samples as validation set.

2.2. Evaluation Metric

As we seek to evaluate event-wise agreement between a given model and the scorers, the metric cannot be computed directly on the segmentation mask. First, events are extracted from the binary mask considering an apnea event is a sequence of consecutive 1. For each apnea event from a window, we extract the start and end index to produce a list of events. This list can be empty if no events are found. The same processing is applied to the ground-truth masks to extract the ground-truth events.

In order to assess the agreement between the ground-truth and estimated events, the F1-score is computed. Two events match if their IoU (intersection over union or Jaccard Index) is above 0.3. Hence a detected event is a True Positive if it matches with a ground-truth event, it is a False Positive otherwise. On the other hand, a ground-truth event without a matching detected event is a False Negative. TP, FP, FN are summed over all the windows to compute the F1-score.

3. Related Work

Previous research conducted on sleep event detection can be divided into two groups: 1) machine learning methods working on hand-crafted features extracted from the raw physiological signals, 2) deep learning approaches leveraging learnt representations of these signals.

Several methods have been proposed for sleep spindle and K-complex detection using a variety of hand-crafted features extracted either in the time domain [3, 4] or in combination with the frequency domain [5, 6]. However, these methods are often event-specific and do not generalize to other types of events. As an alternative, deep learning based methods have been used to automatically learn relevant features from signals.

DOSED [7] is a deep learning based method that aims at jointly predicting locations, durations and types of events in EEG time series data. To do so, it relies on a convolutional neural network which builds a feature representation from raw EEG signals, and two independent modules respectively performing localization and classification of events.

U-Time [8], a fully feed-forward deep learning approach for physiological time series segmentation, was recently introduced for the analysis of sleep data. It is a fully convolutional network inspired by the U-Net [9] architecture (originally proposed for image segmentation) which maps sequential inputs to sequences of class labels on a freely chosen temporal scale. This is done by implicitly classifying every individual time-point of the input signal and aggregating these classifications over fixed intervals to form the final predictions.

RED [10] is a recent deep learning approach for sleep event detection from EEG signals based on convolutional and recurrent neural networks. It leverages two input representations: the raw time-domain EEG signal, as well as a complex spectrogram of the signal obtained via Continuous Wavelet Transform.

4. Working with Fourier Features

4.1. Logistic Regression and SVM

After a first glimpse at the PSG signals and corresponding binary masks provided for the competition, we can make an initial assumption: apnea events have a direct (but not systematic) impact on some of the signals (such as AirFlow, AbdoBelt and ThorBelt for instance). This means that we have some hope to detect apnea events on a window of length 1 second by considering the time series (of length $l = 100$ because of the sampling frequency $f_s = 100Hz$) of those signals for the corresponding window. Nonetheless, proceeding this way, we see that the dimension of the features we could base a classification algorithm on is very high. A way to encapsulate a change of trend or an altered respiratory rhythm is naturally computing the Fourier coefficients of those windows, truncated at a given order $K \in \mathbb{N}^*$. This approach is in the wake of the advice given during the course, so as to reduce the dimension of the features, keeping the information we assume to be more relevant.

This justifies our first attempt: we consider as data set $\mathcal{D} = (X_i, y_i)_{1 \leq i \leq N}$ where $N = 4400 \times 90$ is the total amount of seconds in the PSG signals (thus each sample corresponds to one second), and where for any $i \in [N]$, X_i is defined as the concatenation of the real and imaginary parts of the first K Fourier coefficients of each of the $N_s = 8$ signals after a scaling (subtracting the mean and dividing by the standard deviation per signal). Therefore, for any $i \in [N]$, $X_i \in \mathbb{R}^{2 \times K \times N_s}$ and $y_i \in \{0, 1\}$ is the binary variable equal to 1 if the second corresponds to an apnea event, 0 otherwise.

In this context, we implement two classifiers: a Logistic Regression and a SVM using scikit-learn package. Because of the natural regularization induced by the SVM model, we focus on this latter and proceed to a tuning, testing several kernels (linear, Gaussian, polynomial) and corresponding regularization parameter λ, as studied in details in the optimization course by Alexandre d'Aspremont based on [11].

Either the Logistic Regression or the SVM lead to unsatisfying predictions on our specific task after proceeding to a train-test-val split of the N points of the data set - dealing with the sample imbalance (between 0 and 1 labels) by drawing a balanced sub data set from $\mathcal{D}$ - and after tuning hyper parameters:

- the amount of Fourier coefficients K.
- the choice of the kernel for the SVM model and of the corresponding parameters (order of the polynomial functions, standard deviation of the Gaussian function).
- the regularization parameter λ of the SVM.
- the choice of a subset of the N_s signals of cardinal $p \in [N_s]$, leading to a feature space of dimension $2 \times K \times p$.
- the choice of a representation based on the modulus of the complex Fourier coefficients instead of the real and imaginary parts, leading to a dimension divided by 2 for the feature space.

We compute the confusion matrix encapsulating True positive (TP), False positive (FP), True negative (TN) and False negative (FN) over the validation set and assess a bad performance of around 0.55 accuracy over the data set $\mathcal{D}_{val} \subset \mathcal{D}$. This score is an accuracy for second per second classification, not the metric involved in the evaluation of our performances.

Though this result is not thrilling, we emphasise a first simple and hand-crafted approach to address the task of the competition, which enables us to decouple the representation part from the training and optimizing part of our algorithm - which is often not the case of the state of the art deep learning models in [12]. The main lesson we learn from this experience is that annotated seconds can not be considered as independent from each others. Indeed, the context of a window is crucial. A simple example to illustrate this is the drop of SPO$_2$ visible a few seconds after an apnea (thus introducing a delay) as visible on Figure 1.

4.2. Hidden Markov Model

The previous subsection has paved the way for a temporal dependency to be taken into account in our model. We propose a Hidden Markov Model to formalize this precisely. This choice is backed by the assessment of the efficiency of those models to address segmentation tasks on time series (state of the art before the appearance of RNNs). In this context, the data set is now a set of sequences denoted by $\mathcal{D} = (X_i, Z_i)_{1 \le i \le N}$ where now $N = 4400$ is the amount of samples in the original data set, where for any $i \in [N]$, $Z_i \in \{0,1\}^T$ and $X_i \in \mathbb{R}^{d \times T}$, and where $T = 90$. The main assumptions we make are the following:

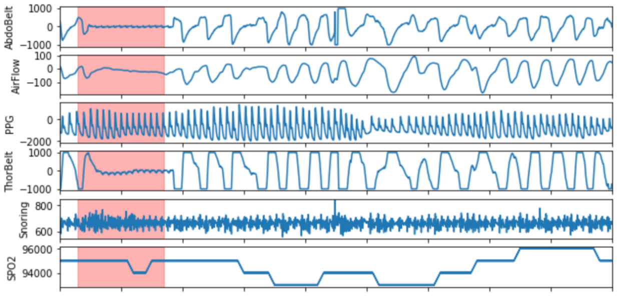

Figure 1: A Sleep Apnea Event: SPO$_2$ drops after the event, whereas AirFlow and Belt movements are instantaneously affected.

- for any $i \in [N]$, $Z_i = (Z_{i,t})_{1 \le t \le T}$ is a realization of a (hidden) Markov Chain with states $\{0,1\}$ and transition matrix $P \in \mathbb{R}^{2\times 2}$, corresponding to the annotations of apnea and non-apnea events.
- for any $i \in [N]$ $X_i = (X_{i,t})_{1 \le t \le T}$ corresponds to the observed realization of the Hidden Markov Model, and for any $t \in [T]$, $X_{i,t} \in \mathbb{R}^d$, $d = 2 \times K \times p$ where as previously $K \in \mathbb{N}^*$ is the maximal order of Fourier coefficients taken to represent a window of one second, $p \in [N_s]$ is the amount of signals considered.
- we assume a Gaussian law for the conditional dependencies: for any $i \in [N]$ and any $t \in [T]$, $p(X_{i,t} = x | Z_{i,t} = z) := \mathcal{N}(x; \mu_z, \Sigma)$, where $\mathcal{N}(x; \mu_z, \Sigma)$ is the Normal distribution of dimension d with mean μ_z (depending on the value of z) an covariance matrix Σ (independent on z) evaluated in $x \in \mathbb{R}^d$.

Then, given the probabilistic model described in details by Bishop in [13], we can learn the parameters (with EM algorithm for instance) on a train set and predict (with Viterbi algorithm and dynamic programming) the sequence of hidden variables Z for a given series of observed variables X in the test set. We suggest an implementation with the `hmmlearn` package. As for the case of simple second per second classification, we do not manage to get interesting results with a train-val-test-split process. We assume this is due to the structure of the inference graph (with Markov assumption over hidden states and conditionals for the observed variables). We thus do no spare more time on this model, though extensions could be considered with a Bayesian approach and a more complex graphical model with other priors.

4.3. Long Short Term Memory Network

The last attempt we make on the hand-crafted Fourier features extracted from the raw PSG signals is a bidirectional Long Short Term Memory (LSTM) network to model the time dependency over each 90 seconds window. The data set we consider $\mathcal{D}$ is the same as in section 4.2. We use PyTorch to build a two layer deep bidirectional LSTM followed by a linear layer with sigmoid activation to project each hidden state of the LSTM's output sequence onto a single probability of event. We address data imbalance with the creation of a balanced sub-data set. We also suggest two ideas: we **add weights to the BCE loss**, with higher values in the locations of ground truth apnea events, and we proceed to a "smoothing" of the $0-1$ labels to induce a structure within apnea events.

After a train-val-test split of the data set and a tuning of the parameters involved (such as the dimension of the hidden states, the amount of Fourier coefficients K considered, the coefficient of the smoothing method, the weights for the loss etc.), the best F1-score we achieve on our test set is 0.24. This is our first attempt with a relevant information drawn from a model on Fourier features, but this score is way below the benchmark provided for the competition. At this step, we choose to switch to deep learning models on raw signals, to draw as much meaning as possible from the PSG. The main difference is that the representation of our data set is no more decoupled from the optimization and learning parts, making the interpretation of what "works" less clear, but avoiding the loss of information induced by a hand-crafted processing.

Configuration	F1-score
SVM	$\ll 1$
Hidden Markov Model	$\ll 1$
2-layer deep + bidirectional LSTM	**0.241**

Table 1: Summary of Results on Fourier Features

5. Deep Learning Models

5.1. Working on a Single Signal

For the sake of simplicity, we decide to start with models designed to process only one of the eight signals available. Each signal can be viewed as a 90s long sequence of 100-dimensional vectors. When it comes to dealing with sequential data such as time series data, Recurrent Neural Networks (RNNs) [14] are one of the most popular architectures. They are particularly convenient as they can process inputs of any length and their size does not increase for longer inputs. Through the use of hidden states and shared weights, they are designed such that the computation for step t can use information from many steps back (at least in theory). In order to produce binary segmentation masks, we stack a linear layer with sigmoid activation on top of three RNN layers: this linear layer projects the hidden state of each time step $t = 1, \ldots 90$ into a single probability of event. We train this model on each input signal independently. Results are summarized in Table 2.

Signal	F1-score
Abdominal belt	0.236
Airflow	**0.278**
PPG	0.136
Thoracic belt	0.243
Snoring indicator	0.123
SPO$_2$	0.232
C4-A1	0.076
O2-A1	0.115

Table 2: 3-layer deep RNN on each Signal

These results suggest that airflow is the signal from which it is easier to predict sleep apnea. The rest of the experiments presented in this section are conducted on this signal.

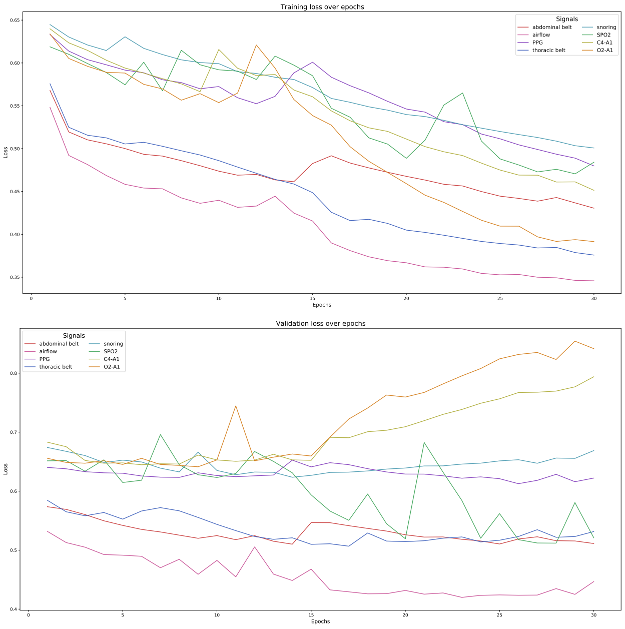

Figure 2: Loss for each signal

In practice, however, RNNs suffer from long term dependencies issues: the network has difficulties accessing information many steps back. This usually causes the

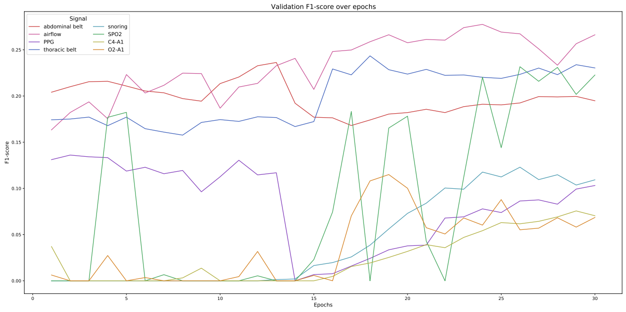

Figure 3: Validation F1-score for each signal

gradient to vanish when back-propagating through earlier layers and the network to "forget" about past information. Quick visual inspection of signals suggests that long-term temporal structure may provide useful information for detection of sleep apnea event. Hence, modeling long-term temporal context should improve detection performances. Several extensions of the classic RNN architecture have been designed to increase the memory capacity of the network along with the features extraction capacity. We decide to explore one of them: Long Short-Term Memory networks.

LSTMs are a type of recurrent neural networks that allows long-term dependencies in a sequence to persist in the network by using "forget" and "update" gates. At each step *t*, we do not only have a hidden state h_t, but also a cell state c_t, which stores long-term information. LSTMs can erase, write and read information from the cell: the selection of which information is erased / written / read is controlled by three corresponding dynamic[1] gates. The cell acts as a route allowing the gradient to flow through the network without vanishing and solves the long term dependencies issues that plain RNNs suffer from. Switching from standard RNN to LSTM slightly improve the F1-score when working on airflow (from 0.278 to 0.288, Table 3).

We can regard the hidden state h_t corresponding to the vector of values at a given time step *t* as the representation of this vector in the context of the 90s long signal. In the case of a plain LSTM, this contextualized representation only contains information from the *left* context. Yet, the *right* context, which corresponds to the signal's values following the current time step *t* may change the interpretation we have from this vector. This is precisely what motivated the development of bidirectional LSTMs, which allow to have information from both left and right when computing the contextualized representations of vectors in a sequence. Switching from unidirectional to bidirectional nearly doubles the F1-score when working on airflow (from 0.288 to 0.561, Table 3).

Configuration	F1-score
3-layer deep RNN	0.278
3-layer deep LSTM	0.288
3-layer deep + bidirectional LSTM	**0.561**

Table 3: RNN vs. LSTM Results on Airflow Signal

Inspired by previous work [7, 8, 10], we then decide to use 1-dimensional convolutions to extract useful features from raw physiological signals. The idea would be to use these features to directly predict sleep apnea events, or as pre-processed input to a recurrent neural network. In order to predict sleep apnea events from convolutional features, we stack three 1-dim convolutional layers and add on top of them a linear layer with sigmoid activation to project each convolutional feature onto a single probability of event. We train this model on each signal independently. None of these experiments leads to a F1-score worth reporting. We hope that stacking a bidirectional LSTM on top of these 1-dimensional convolutions and training the whole model end to end can capture the long-term dependencies on learnt representations. Surprisingly, it does not give better F1-scores.

Working on a single signal gives interesting results. But to go one step further, we have to come up with a novel training strategy and architecture to process multiple signals together.

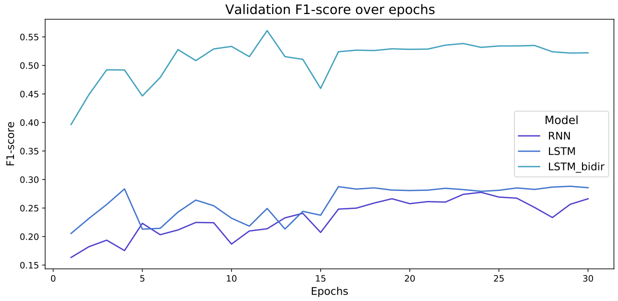

Figure 4: Validation F1-score of different models on airflow

5.2. Working on Multiple Signals

For each sample in the data set, the eight signals available can be modeled as a tensor in $\mathbb{R}^{8\times 90\times 100}$. Each signal may hold important information for predicting sleep apnea events over the 90s sequence. But this information is not necessarily synchronized between signals: visual

[1] dynamic means their value is computed based on the current context

inspections of samples suggest the Airflow is disturbed during the apnea event, while the O_2 saturation of the blood (SPO$_2$ signal) often drops once the apnea event has ended. Based on this observation, we conclude that whenever we use convolutions, it does not make sense to convolve signals together: convolutional filters learnt at a given time step may mask important information at another time step.

Once again, our first approach to this problem consists in concatenating the raw 100-dim vectors of the selected signals at each time step t. For example, when using the eight signals together, this would give 800-dim vectors at each step $t = 1, \dots, 90$. We feed these sequences of concatenated input vectors to a bidirectional, 3-layer deep LSTM, trying different combinations of signals. For each experiment, we adapt the hidden size of the LSTM as the closest power of 2 to the size of input vectors. Unfortunately, none of the various combinations we try proves to be successful.

Then, in the same fashion as in Section 5.1, we decide to pass our signals through 2-dimensional convolutions in order to extract useful features. As stated above, it does not really make sense to share convolution kernels between different signals. Hence, we use grouped convolutions to process each signal independently. Our backbone architecture consists in three 2-dim grouped convolutions, followed by one 1×1 grouped convolution to compress the feature maps into a single channel. When processing $N \leq 8$ different signals together, this would map the input tensor from $\mathbb{R}^{N\times90\times100}$ to $\mathbb{R}^{N\times90\times30}$. We then either add a linear layer with sigmoid activation to project each convolutional feature onto a single probability of event: from $\mathbb{R}^{N\times90\times30}$ to $\mathbb{R}^{N\times90\times1}$ (*GroupedConv2d* model), or concatenate these 30-dim vectors signal-wise to produce $N \times 30$-dim vectors later fed in a 3-layer deep, bidirectional LSTM (model).

We try processing different subsets of signals. The best performances is achieved when combining the abdominal belt, airflow, thoracic belt and SPO$_2$ signals. Interestingly, these are the four signals that gives the best F1-score when processed alone, as shown in Table 2.

Model	F1-score
GroupedConv2d	0.072
GroupedConv2d + LSTM	**0.546**

Table 4: Results on abdominal belt + airflow + thoracic belt + SPO$_2$ signals

Eventually, using a bidirectional LSTM on airflow or using the more sophisticated *GroupedConv2d+LSTM* gives similar results (Figure 5). We hoped integrating more signals would have helped us gain in performances, but this is not the case. We still believe there is a strong potential in combining multiple signals together, and briefly detail some ideas in Section 6 we come up with as perspectives.

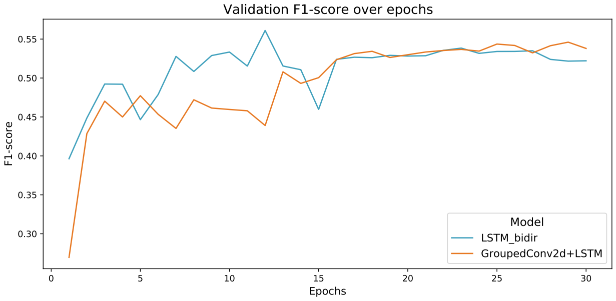

Figure 5: Comparison of the two best performing models over epochs

5.3. Addressing the specificity of the data set

After a few iterations and attempts dealing with the provided data set, we are aware of some of its characteristics:

- It is heavily unbalanced: apnea labels represent 6.87% in the 4400×90 seconds groundtruth segmentation masks.
- No structure is present in the label of apnea events, that is to say short apnea events are denoted by 1s labels, so are long ones, and being at the middle of an apnea window does not influence this label compared to being at the beginning or at the end.
- Apnea events are supposed to last more than 10s and are characterized by connected windows of 1s in the labels. The classical BCE loss does not take this crucial point into account.

Therefore, we implement some ideas to deal with the latter points:

- As in section 4.3, we introduce some weights in the BCE loss, penalizing the predictions over ground truth apnea events heavier. A factor of 3 is chosen after tuning.
- We also introduce a smooth representation of the labels with a Gaussian function applied on apnea events, centered at the middle of each window and having a standard deviation tuned: $\sigma = 2 \times l$ where l is the length of a given apnea event.
- To force connected predictions, we define a new Pytorch class working as follows: for each predicted vector $\hat{y} \in [0, 1]^{90}$, we compute the average of the probabilities in a window centered at every second having a

fixed length l_c leading to a new vector $\hat{y}^{avg} \in [0,1]^{90}$. Then, for any $1 \leq t \leq 90$, we switch the predicted label if $\hat{y}_t^{avg} \leq \tau_{\text{low}}$ and $\hat{y}_t$ leads to apnea prediction, or if $\hat{y}_t^{avg} \geq \tau_{\text{high}}$ and $\hat{y}_t$ leads to absence of apnea prediction. We tune the parameters l_c, τ_{high} and τ_{low}, and we use this function either during training or as a post-processing. For our final submission, we use $l_c = 11$, $\tau_{\text{low}} = 0.1$ and $\tau_{\text{high}} = 0.5$ only as post-processing. Note that to get a binary class from a probability in $[0,1]$, we found the best working classification threshold to be $\tau_{\text{class}} = 0.7$.

All those ideas enable to improve our score by a significant amount, and results in the previous sections include their use after tuning.

6. Discussion

We notice that the attempts to predict apnea events on Fourier features do not lead to the best results. Some comments can be made about this assessment: since Fourier features extracted in section 4 encapsulate the local information of each second, they do not include any context. An idea to extend this hand-crafted work would be to consider surroundings of seconds to compute Fourier coefficients, or to use other descriptors such as wavelet coefficients. Besides, the second-per-second and the HMM models are intrinsically limited by the assumptions they are based on. As evoked in section 4.2, a way to go further in the direction of non-deep learning models would be to define a more expressive probabilistic model. This could be a way to introduce some uncertainty in the predictions, not only focusing on the Maximum A Posteriori (MAP) of the weights of the models but also on their posterior distribution. This is particularly useful to avoid over-fitting and to introduce prior knowledge as it is the case in the medical field. Because of the deadline of this competition, we had no time to investigate those alternatives. The main asset of section 4 is to study models whereby data representations and predictions are - at least partially - decoupled.

When working with deep learning architectures on raw signals (Section 5), we implicitly represent our dataset in a low dimensional space, either using convolutions or focusing on the hidden states of recurrent neural networks. This representation is learnt during training and applied to new PSG samples for time series segmentation. It is crucial to foster a proper generalization on a new data set. Through iterations with different models, architectures and parameters, we experience a less precise knowledge than with “simpler” models of the best practices because of the intrinsic interdependence of the phases of the end-to-end models. We highlight the crucial need to address the specificity of the dataset in section 5.3. Furthermore, we observe better results using a single channel (AirFlow) than taking into account the whole PSG. This might be due to the need for a deeper reflection on the way to combine the information of the channels. Though we suggest a first way to do so in section 5.2, some perspectives could be considered, either based on different convolutional architectures to feed a LSTM - such as the idea of an ICA computation described in [7] - or involving some data specific processing. For instance, we see that the meaningful representation of the O_2 saturation signal seems to be its mean value over seconds. This feature has a drastically lower dimension than the representation created by a CNN for instance, but the encapsulated information seems crucial as noted in [1]. Therefore, stacking it to a higher dimensional vector does not seem to be the best choice, and treating it as other signals could entail over-fitting because of the unnecessary high dimensional representation induced.

An information we could consider in a further study would be the patient id. Indeed, as stated in Section 1, sleep apnea can be either central or obstructive. For central apnea, the thoracic belt and abdominal belt signals are discriminatory factors whereas for obstructive apnea they might be misleading. Hence, we believe a preliminary unsupervised classification of patients (using clustering methods for instance) could highlight different sleep apnea profiles, for which we could train separated models. This would reduce the amount of data each model is trained on, but as we expect the patients among a given profile to behave similarly, using lighter models could overcome this limitation.

Eventually, we highlight two possible avenues to investigate for future work. First, modelling sleep apnea detection as a multi-label classification task limits the results we can expect to get from the data at hand. Despite being relatively easy to implement, we believe modeling the problem as a detection or a segmentation task should lead to improvements. Then, we tried experimenting with a Transformer [15], an attention-based architecture for modeling sequential information that is an alternative to recurrent neural networks. Unfortunately, we didn’t have enough time to correctly study this architecture. Not only it could bring performances to the next level, but through in-depth analysis of its attention mechanism, it could lead to interpretable outputs, cracking the deep learning *black box* open.

7. Conclusion

In this report, we highlight the main process we have chosen to address the Dreem challenge for the MVA course Sparse Representations. An emphasis over the iterations we followed for the assumptions, representations of the data set and models enables us to pinpoint the main challenges to be able to segment PSG signals, with unbalanced labels and diluted information in relatively high-dimensional raw features. Our last model has reached a 0.6503 academic private score. It could be improved using the perspectives described in Section 6.

References

[1] Karl A. Franklin and Eva Lindberg. Obstructive sleep apnea is a common disorder in the population—a review on the epidemiology of sleep apnea. 7(8):1311–1322. 1, 7

[2] Sheikh Shanawaz Mostafa, Fábio Mendonça, Antonio G. Ravelo-García, and Fernando Morgado-Dias. A systematic review of detecting sleep apnea using deep learning. 1

[3] Ankit Parekh, Ivan Selesnick, Ricardo Osorio, Andrew Varga, David Rapoport, and Indu Ayappa. Multichannel sleep spindle detection using sparse low-rank optimization. *Journal of neuroscience methods*, 288:1–16, 08 2017. 2

[4] J LaRocco, P J Franaszczuk, S Kerick, and K Robbins. Spindler: a framework for parametric analysis and detection of spindles in EEG with application to sleep spindles. *Journal of Neural Engineering*, 15(6):066015, sep 2018. 2

[5] Karine Lacourse, Jacques Delfrate, Julien Beaudry, Paul Peppard, and Simon C. Warby. A sleep spindle detection algorithm that emulates human expert spindle scoring. *Journal of Neuroscience Methods*, 316:3–11, 2019. Methods and models in sleep research: A Tribute to Vincenzo Crunelli. 2

[6] Daniel Lachner-Piza, Nino Epitashvili, Andreas Schulze-Bonhage, Thomas Stieglitz, Julia Jacobs, and Matthias Dümpelmann. A single channel sleep-spindle detector based on multivariate classification of eeg epochs: Mussdet. *Journal of Neuroscience Methods*, 297, 12 2017. 2

[7] Stanislas Chambon, Valentin Thorey, Pierrick J. Arnal, Emmanuel Mignot, and Alexandre Gramfort. DOSED: a deep learning approach to detect multiple sleep micro-events in EEG signal. 2, 5, 7

[8] Mathias Perslev, Michael Jensen, Sune Darkner, Poul Jørgen Jennum, and Christian Igel. U-time: A fully convolutional network for time series segmentation applied to sleep staging. page 12. 2, 5

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 2

[10] Nicolas I. Tapia and Pablo A. Estevez. Red: Deep recurrent neural networks for sleep eeg event detection. *2020 International Joint Conference on Neural Networks (IJCNN)*, Jul 2020. 2, 5

[11] Stephen P. Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press. 3

[12] 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)*, pages 1596–1600. ISSN: 1558-4615. 3

[13] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer. 3

[14] D. Rumelhart, Geoffrey E. Hinton, and R. J. Williams. Learning internal representations by error propagation. 1986. 4

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. 7