Application of modern machine learning techniques to the detection of micro-awakenings on a midsagittal jaw motion signal

Auteur : Hockers, Pierre
Promoteur(s) : Sacré, Pierre; Geurts, Pierre
Faculté : Faculté des Sciences appliquées
Diplôme : Master en ingénieur civil en informatique, à finalité spécialisée en "intelligent systems"
Année académique : 2020-2021
URI/URL : http://hdl.handle.net/2268.2/11239

University of Liège
Faculty of Applied Sciences
Academic year 2020-2021

Application of modern machine learning techniques to the detection of micro-awakenings on a midsagittal jaw motion signal.

A study submitted in partial fulfillment of the requirements for the degree of Master in Computer Science and Engineering

by **Pierre Hockers**

Advisors :

Bernard Beckers - Nomics S.A
Pierre Geurts - University of Liège
Pierre Sacré - University of Liège

Abstract

Sleep apneas are common diseases that impact the quality of our lives. It is estimated that between 0.5% and 5% of the population suffers from it. The most common apneas are obstructive, which means that they are caused by respiratory efforts, but they can also be caused by neurological problems impacting the respiration process. These disorders can be the source of many subsequent health problems, which makes their detection useful.

Nomics is a company which introduced a novel sensor able to detect the breathing patterns associated with these disorders. As some previous work showed [1], this sensor could allow automatic detection of sleep apneas, in a more convenient fashion than currently performed.

This master thesis is concerned with the use of modern machine learning techniques to tackle the challenge of automatic detection of the arousals associated with these sleeping disorder breathings. We will start by a review of the current procedure used by doctors, then identify the limitations of the method proposed in Dr Senny's PhD thesis [1] before building a data pipeline to create datasets we can use to train Random Forests, fully connected Deep Neural Networks and Convolutionnal Neural Networks to detect these patterns. We detail the model that obtained our best performance and discuss our final results.

Acknowledgments

Je dédie ce mémoire à mes grands parents, en particulier à Papou, en mémoire de ses études.

J'aimerai remercier plusieurs personnes qui m'ont aidés à un moment ou l'autre durant ce projet.

D'abord, merci à mes amis, Thomas Romain, Olivier et Simon.

Thanks to Gjiltina as well to make the end of my adventure at the university a lot sweeter.

Ensuite, merci à mes parents pour leur soutien tout au long de ma scolarité. Vous avez été parfaits.

Mais mes plus grands remerciements vont à mes encadrants, Mr.Geurts, Mr.Sacré et Mr. Beckers, qui ont su m'aider et me débloquer quand je ne savais plus quoi faire. Je n'ai jamais autant appris qu'en les regardant réfléchir avec moi sur les différents problèmes qui se sont posés. Un merci tout particulier à Mr. Beckers qui était toujours disponible pour discuter avec moi du sujet, et pour répondre à mes innombrables questions.

Contents

Chapter 1

Introduction

If we had to point to the activity humans spend the most time doing, the answer would probably be sleep. Sleep takes up to a third of our lives, and plays a primordial role in our well being, our health and our ability to perform other tasks. Therefore, having a good quality of sleep is a necessity.

Unfortunately, sleep can be disturbed by many events, some of which come from respiratory issues. These are called sleep disorders breathing (SDB), and can cause poor sleep quality by hurting its resting benefits, which in turn can lead to fatigue, irritability, depression, cardio vascular issues, and, in some rare cases, death. These disorders can be ramified in various subcategories, such as apneas, respiratory efforts and hypopneas. They are often followed by a short event called a micro-awakening, caused by a response of the nervous system to the breathing difficulties experienced. Because of this, they are also called arousals. These micro awakenings fracture sleep and impact the healing process of the sleep.
A patient suspecting something to be wrong with his sleep can, luckily, get help. Indeed, four physiological and measurable parameters allow a skilled practician to identify these sleeping disorders : nasal airflow, oximetry, an arousal marker, and a respiratory effort marker. However, the tools and environment needed to perform such a diagnostic are impractical, as they are intrusive to the patient. The potential sufferer indeed needs to spend a night in a sleep clinic, to pass the procedural sleep examination : the polysomnography (PSG). This procedure requires the patient to sleep with a battery of sensors recording various signals and physiological metrics. The combination of these intrusive captors and the general discomfort of sleeping outside of one's home can impact the viability of the results. Manually scoring the PSG is also a long and tedious task, and sleep clinics typically have limited beds and personal.

The company Nomics has developed a sensor called Jawac that measures the jaw movements and can be used as a substitute to measure 3 of the 4 aforementioned physiological parameters neeed to diagnose SDBs. These qualities make the Jawac sensor a good potential alternative to the classic procedure for the gathering of sleep analysis signals, and is easy enough to use so that a patient could take the sensor home and record his sleep on his own before bringing back the results for analysis.

Eleven years ago, a PhD thesis [1] was written by Frederic Senny to explore the viability of this new signal. As a part of this thesis, he then applied simple algorithms to the Jawac signal to try to detect micro-awakenings. While his approach proved the potential of the method, it still showed limitations.
The field of Machine Learning has since then evolved, and powerful techniques like Deep Learning [2] have since proven to be able to tackle difficult classification tasks [3]

In this thesis, we will explore modern machine learning methods applied to the detection of micro awakenings in a Jawac signal, in an effort to automatize the detection of patients victim of breathing sleeping disorders. We will build a data pipeline for the processing of the data at hand, and will then explore various modern algorithms, analyse their results, and try to improve over the performances of Dr Senny's algorithm.

The structure of this report is the following :

- We start by introducing the necessary physiological background on the subject at hands in Chapter 2
- Chapter 3 is dedicated to a brief overview of basic machine learning definitions, algorithms and relevant concerns to this thesis.
- Chapter 4 reviews state of the art approaches on the subject of sleep breathing disorders detection with machine learning techniques as well as some previous work on the Jawac.
- Chapter 5 details the pipeline we built to transform our raw data into a usable dataset.
- Chapter 6 presents the various models used and their best version each, before presenting the overall best model.
- Chapter 8 discusses the results obtained in Chapter 7
- Finally chapter 9 presents our conclusions and displays a list of suggestions for future work on the subject.

Chapter 2

Physiological context

2.1 Sleep-disordered breathing and micro awakenings

2.1.1 Sleep Disorders Breathing

Sleep-disorder breathing is a family of pathologies that are caused by the partial or total collapse of the upper airways. They can be divided in three main categories : hypopneas, apneas and respiratory effort related arousal (RERA).
As an insight, 4% of the adult males of Western countries and 2% of females will suffer from obstructive apneas or hypopneas [5]. We will here define each of them briefly.

Apneas

A sleep apnea is characterized by an interruption of the breathing process while sleeping . A more clinical definition of an apnea is a reduction in the nasal airflow of more than 90% of the original airflow lasting for at least 10 seconds.

There exists 3 subcategories of apneas, differentiated on the absence, presence or partial presence of respiratory effort, but all characterized by a cessation of the naso buccal airflow of at least 10 seconds:

- Central Apnea : lack of central respiratory drive
- Obstructive Apnea : persistence of ventilatory effort during apnea, due to occlusion of the upper airways.
- Mixed apneas : hybrid result of both previous cases. Apnea that starts with a central apnea and ends with an obstructive.

Hypopneas

Hypopneas are defined with a reduction of airflow of at least 50% or a reduction of 30% of the original value associated with a loss of at least 3% in the arterial blood oxygen saturation or an arousal [6].

RERA

As indicated by its name, the Respiratory Effort Related Arousal is a sequence of breaths inducing efforts and followed by a micro awakening (arousal) that does not match apneas definitions.

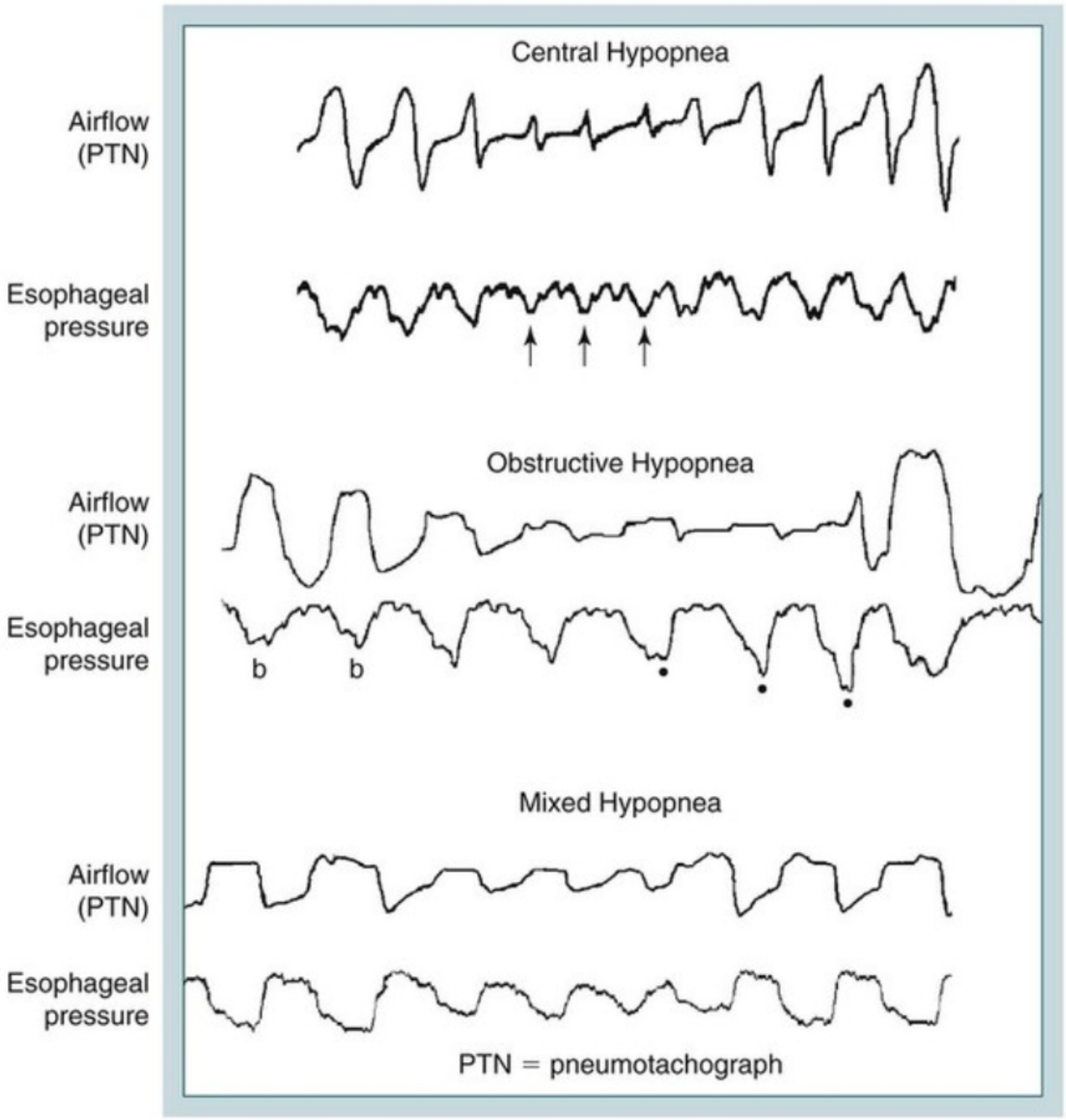

Figure 2.1: Visualisation of Sleep Disorders Breathings symptoms [4]

Impact of SBDs

SBD can cause severe problems such as cardiovascular issues, diabetes, impact the psychologic health, and more. In some case, it can be lethal.
SBD can cause the brain to react to the disturbance, inducing a micro awakening, a brief arousal. Arousals are the nervous system answer to the stimulii that SDBs are and fracture sleep. These are unconscious, making it a very insidious pathology, as the night appears to have been complete and uninterrupted to the patient, who will nonetheless suffer the symptoms.

2.1.2 Arousals

Arousals are small transient awakenings that do not result in behavioral awakening. They are detrimental to sleep quality as they fragment it, leading to daily sleepiness. It is however theorized by Peter Halaász et al. that arousals could also structure sleep by assuring the reversability of sleep, preventing one to fall in what would otherwise be similar to a coma [7].

Classic arousing stimulii encompass various sleep breathing disorders that can be identified, thanks to analysis of electroencephalograph (EEG) and/or polysomnographs (PSG). The ASDA (American Sleep Disorder Association) defines arousals as "[...] abrupt shifts in EEG frequencies [...]", that can be identified by a set of criterias mostly based on the behavior of the EEG frequencies and several other physiological measurements [8]. An arousal can however only occur after a continuous 10 seconds of sleep period, and a second arousal cannot be scored unless another period of 10 seconds of sleep occurred in between.

Most of the SDBs are associated with arousals, which as mentionned earlier appear on the EEG. This indicates that the nature of the arousals we are concerned about in the work is a response from the nervous system to a stimulus (here a SDB). These nervous response result in several observable changes of behaviour from the patient, allowing their detection and diagnosis. Their transient nature however makes their detection problematic for the classic 30 second long epoch diagnosis usually performed on PSGs [8].

In this work, we will use the terms arousal and micro awakening interchangeably.

2.2 Electroencephalogram and Polysomnograph : the classic approach

2.2.1 Electroencephalogram

The electroencephalogram (EEG) is a set of signals resulting from non intrusive sensors disposed on the scalp of a patient and measuring the electrical activity of the brain. This signal is one of the additional starting point allowing doctors to score a patient's sleep. It also allows them to build up the hypnogram, an additional signal useful to us to determine the periods where the patient is awake or asleep (and all intermediate stages) and focus only on the asleep parts.

2.2.2 Polysomnography

The Polysomnography (PSG) is the golden standard for the diagnosis of SDBs. It is recorded in a sleep clinic, where the patient needs to sleep while being monitored with all kind of sensors, some of which being intrusive and producing discomfort.
This procedure produces a vast amount of data that then needs to be manually analysed by a sleep expert, a tedious and time consuming task. It is also expensive.
After all these inconveniences, one could legitimately wonder why this PSG is still the golden standard of the field. After all, takeaways monitors with sensors that are easy to place by the patient himself, in his own house, exist and perform. The reasons are simple. Take home monitors only have 4 sensors at best, compared to the 8 the PSG is composed of, and the possibility for an error in the use of the sensors is much more present when no doctor is involved in the process.

2.3 The Jawac sensor : a novel approach

When in need for air, the human body reacts by opening the mouth, to allow better flow. This also applies when we are asleep, and when the lack of air is linked to SDB. This observation led to the design of the Jawac sensor, a sensor measuring the midsagittal jaw movements. In [9], R. Poirrier observed that SDBs were characterized by patterns of the jaw movements, and that respiratory efforts were creating noticeable low frequencies behaviours, confirming the potential of this new idea for the analysis and diagnosis of SDB.
The Jawac sensor is constituted from two metallic coils disposed on the forehead and the chin. It works by generating an magnetic impulse by induction on the emettor coil and measuring the magnitude of the signal received at the receiver coil. Since the magnitude is a function of the distance between the two coils and that the power of the emettor signal is known, this allows for an automatic monitoring of the jaw movements.

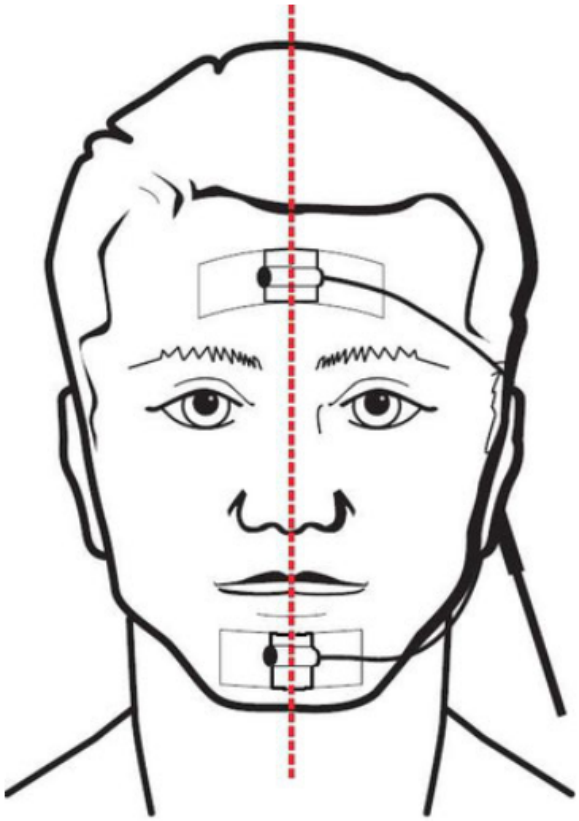

Figure 2.2: The Jawac sensor

2.4 Micro-Awakening detection using the Jawac sensor

The Jawac sensor presents some advantages making it a good candidate for a takeaway sensor for SDBs diagnosis. Only one signal is registered, making it easier to place by a patient alone. It provides enough informations to identify SDBs and arousals (see Figure 2.3), as well as differentiate wake periods from sleeping ones [1, 10] . Moreover, the signal has shown an interpretability automatable by algorithms [1]. This, if made efficient and reliable enough, could make the Jawac sensor a viable alternative to the classic PSG analysis, and bring relief to doctors and sleep clinics that cannot deal with the constant flow of patients by performing a quick and cheap screening to detect suspicious patients that could then be taken in charge more thoroughly by sleep specialists.

While detecting SDB directly is obviously an objective in itself, detecting arousals can be just as useful. Indeed as we mentionned, micro awakenings are often caused by a SDB, as the arousal is the brain way to make the nervous system react to the sudden lack of oxygen. As a matter of fact, the frequency at which someone experiences micro awakening during a night of sleep (the arousal index) correlated with the frequency of apnea and hypopnea in one's sleep [11]. This arousal index has also been shown to be correlated with fatigue [12], one of the classic symptoms of SDBs.

On the basis of all these elements, the goal of this master thesis is to explore modern machine techniques to automatise the detection of arousals on the sole basis of the Jawac sensor, and try to enhance the results precedently obtained by F. Senny in [1].

2.5 Hypnograms, mk3 and additional ressources

To our disposal, we have several additional tools that need to be introduced before hand.

- Hypnogram : the hypnogram is a signal made by doctors on the basis of the EEG. It indicates whether the patient is asleep or not at a given time.This will allow us to focus only on the asleep parts of the Jawac signal
- Various markings have been made over zones of the signals we were provided, indicating various informations about these zones, including the presence of micro awakenings. These markings are provided in mk3 files, a format made for the visualization of these markings on the signal on Nomics' software, Apios. Nomics provided codes making the processing of the markings referenced in mk3 files convenient.
- Predictions made by the algorithm of F. Senny have been registered in mk1 files, a format analog to mk3.
- Apios, the software of Nomics, allowed to visualize the signals provided and the various markings present in the mk3 files. An example is provided in the Figure 2.4, where 10 minutes of the Jawac signal as well as its corresponding hypnogram (marked Hypno) are displayed. The colourful highlights on the "Jawac" signal show marked zones.

MANDIBULAR BEHAVIOUR: During normal sleep the jaw presents no or very little motion, whereas when abnormal breathing efforts occur, the mandible oscillates at the breathing frequency. The mandible lowers during abnormal inspiratory efforts, with an amplitude which is soundly correlated with esophageal pressure (Figure 2).

While abnormal breathing efforts are accompanied by mandibular oscillations at the breathing frequency, arousals that terminate breathing events are recognizable by wide, rapid closing and opening movements of the jaw. These movements appear as "discontinuities" in the Jawac signal.

Moreover, the mandible is subject to different behaviors during different types of abnormal breathing events. Therefore, the analysis of mandibular movements during sleep allows to detect and to classify sleep-disordered breathing events [5].

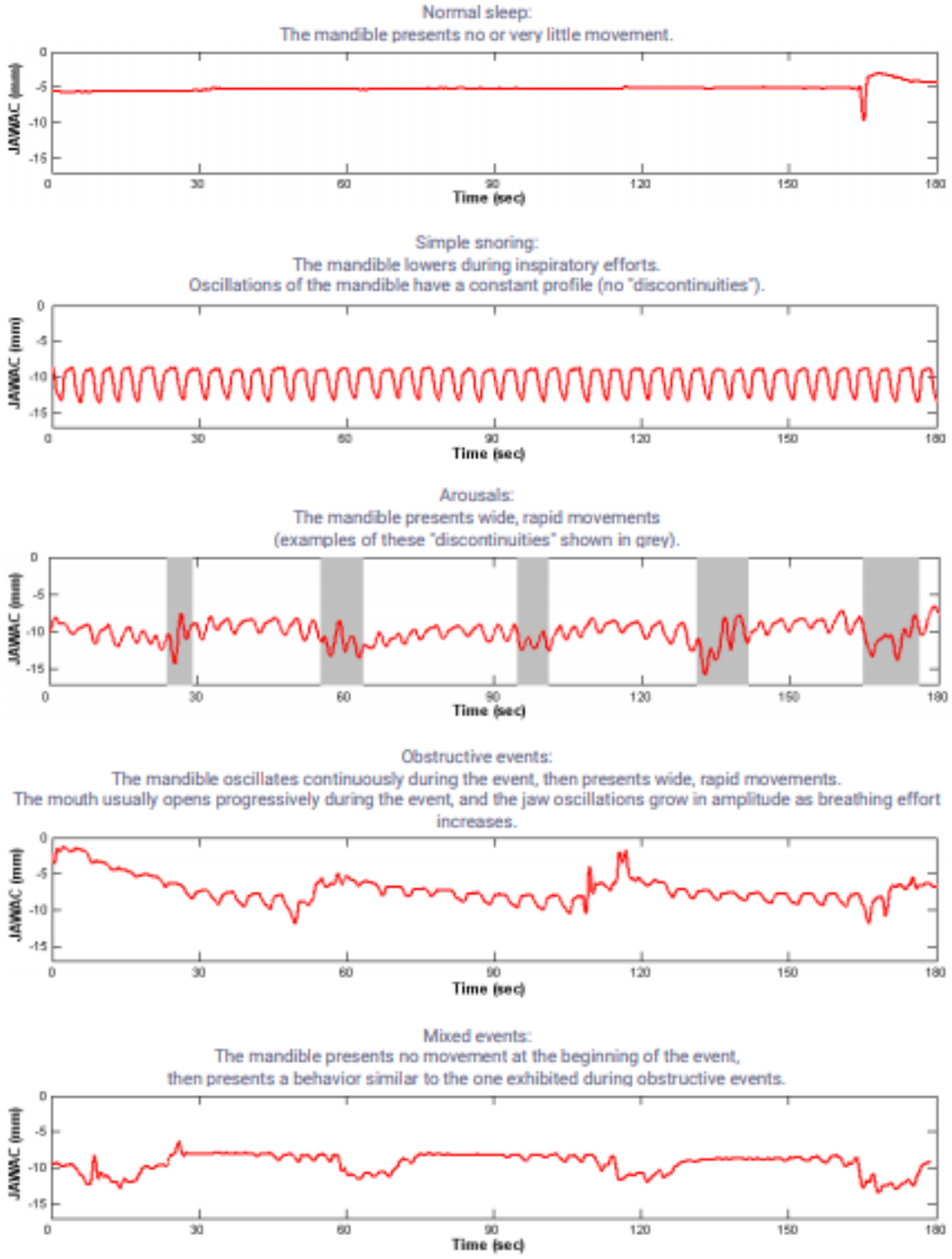

Figure 2.3: SDB visualized on the Jawac signal [13]

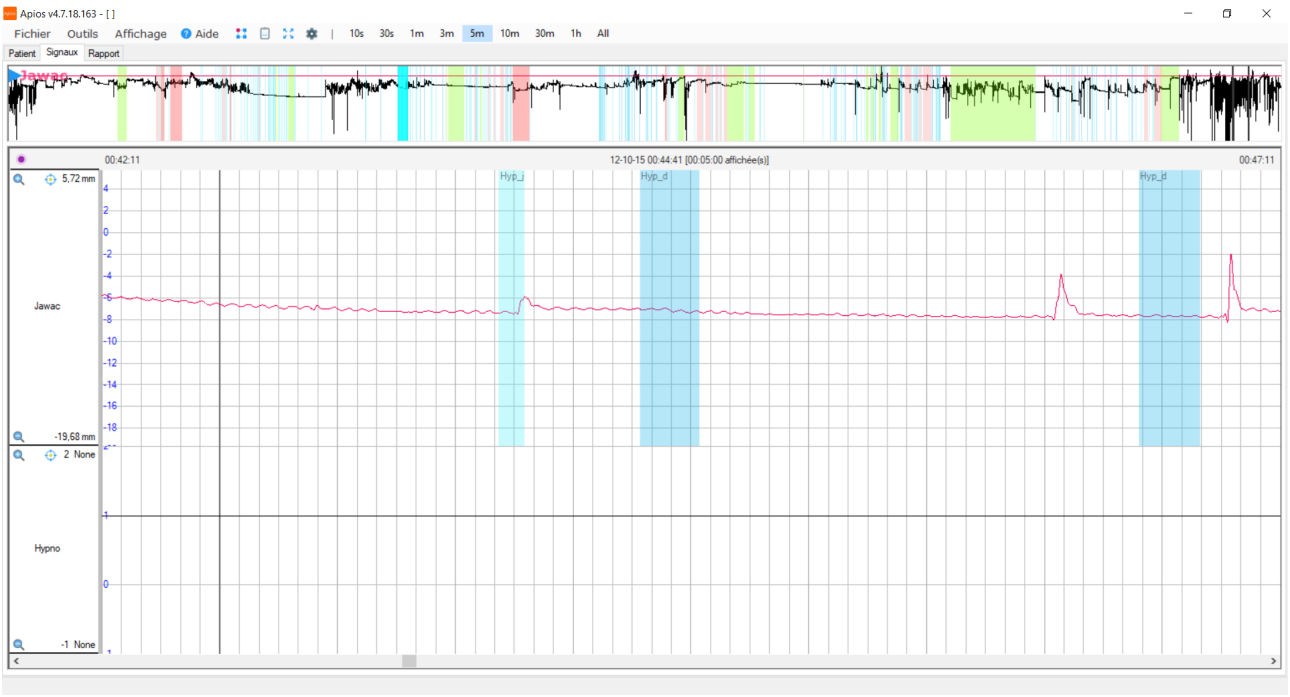

Figure 2.4: Apios visuals

Chapter 3

Supervised Learning

3.1 Formulation of the thesis goal as a supervised learning problem

Supervised learning is a subfield of machine learning concerned with learning an approximation function f which associates an output $\hat{y}$ to a set of inputs variables $\{x_1, x_2, x_3, ..., x_N\}$ from a learning set of input-output pairs. This function is learned from labeled training data representing training examples [14]. This function should minimise the expectation of a loss function $l : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathcal{R}$ over the joint distribution of the input/output pairs $E_{x,y}(l(f(\mathbf{x}), y))$ [15].
The loss function quantifies how far from the actual output Y the function output $\hat{Y}$ is.

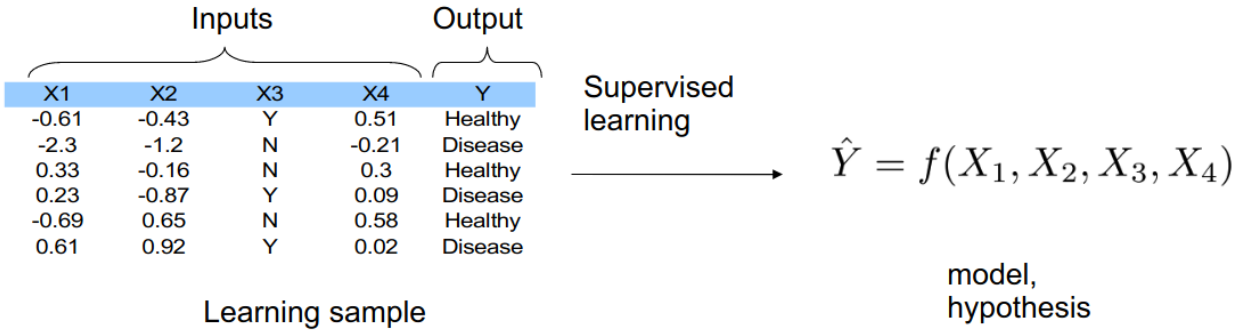

Figure 3.1: Supervised learning aims to learn a function f from a set of learning samples x_i, y_i with $i = 1, .., N$ that approximates the correct output Y by a value $\hat{Y}$ minimising a loss l

The problem we have at hand is an example of supervised learning. Indeed we have signals that we want to associate by zones of 5 seconds to one of two classes : healthy sleep (class 0) and micro-awakening (class 1). Each signal segment of 5 seconds corresponds to a sequence of 50 variables (since the Jawac signal is sampled at 10Hz) associated to a class label.

The loss function used to train the approximation function f will depend on the model itself and can be a hyperparameter in itself.
Similarly, the evaluation function we will use to evaluate models and compare them to each other will differ depending on what we try to evaluate. This will be explained in more details in the "Performances metrics" section.

3.2 Algorithms

3.2.1 Random Forest

While the goal of our work is to experiment with modern machine learning techniques such as Deep Neural Networks on the problem at hand, we have used the popular Random Forest algorithm for the dual purpose of benchmark performances and data viability estimation. Indeed, when looking for an optimal DNN architecture, it is useful to have a base performance to compare to. Furthermore, the data we used was not flawless and some manipulations had to be done before the data was good enough to learn from. Having a simpler model allowed us to notice the moment where data became usable, which allowed us not to waste time on Deep Learning training and network tuning with data in poor shape.

Random Forests are a vastly popular ensemble method based on a group of $N\_estimators$ weak classifiers. The idea is to train these weak classifiers independantly from one another to end up with a group of independant classifiers that can each take a vote on the class to associate to the input. The core idea is that the ensembles of weak learners can make powerful classifiers as they are good at lowering variance without increasing bias and are more robust than single evaluators. These weak classifiers are decision trees [16] in the case of Random Forest[17], as the name implies. Classic Decision Trees tend to overfit,and are therefore prone to variance. An added randomness at two steps of the training of each weak classifier makes the ensemble much more robust against variance. The building process of each of the model's trees is the following :

- build the tree from a bootstrap sample
- When splitting nodes, instead of looking for the best attribute among all of them, look for the best attribute in a random subset of all attributes.

The prediction correspond to the majority of the votes of the weak classifiers.

In this thesis, we used the sklearn implementation [18] of the random forest classifier.

3.2.2 Neural Networks

Basic concept

Neural Networks are a type of supervised learning model that has proven really effective in recent years thanks to the increase in computation power and available data.

They consist in a series of various layers. Each layer is constituted of individual perceptrons, called neurons (illustrated in 3.3). Each neuron takes some inputs values multiplied by a given weight and then add a bias. An activation function is then used on the resulting value and outputs the final value. We distinguish 3 main types of layers :

- Input layers : this is the first layer of the network, in which the inputs values of the neurons are the ones of the input variables.
- Output layers : layers that produce the final prediction.

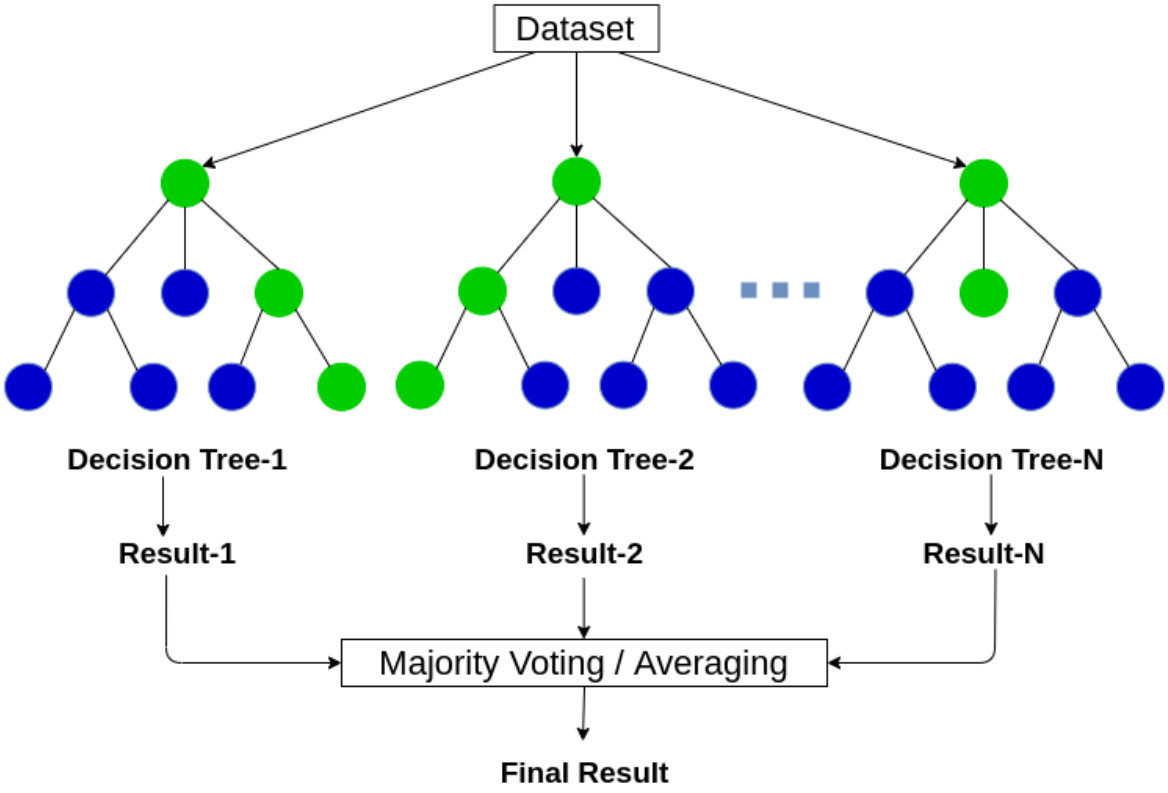

Figure 3.2: Random Forest illustration

- Hidden layers : intermediate layers between the input and output layers. The inputs are the outputs of the previous layers and so on.

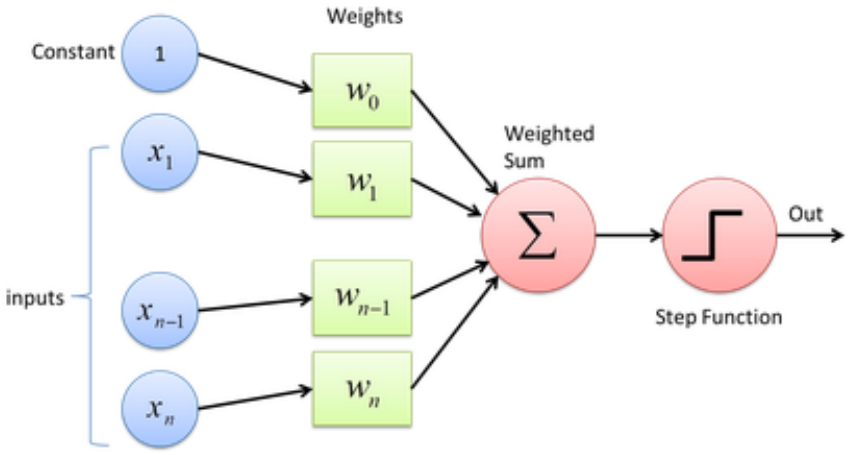

Figure 3.3: Perceptron

Neural Networks are trained on batches of inputs-outputs pairs. A loss is evaluated on these batches. The network uses gradient descent to modify the weights and biases based on a first order differential of the loss function with respect to the weights and biases, efficiently computed using an algorithm called back-propagation. This brings the weights closer to a local minima in the loss function. A hyperparameter called the learning rate dictates the amplitude of the movement in the direction of

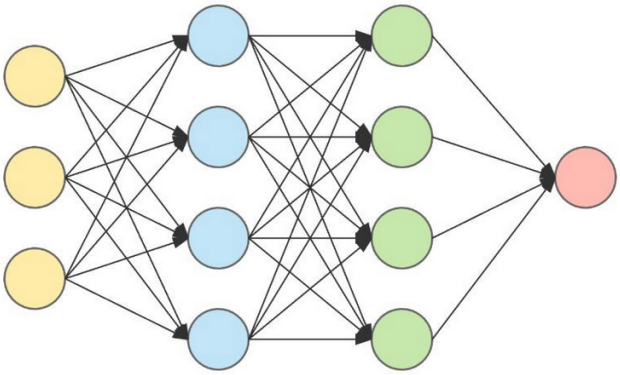

Figure 3.4: A neural network with two hidden layers. In yellow, the input layer. In red, the output layer. In blue and green, the hidden layers.

the minima. This should be carefully defined as a too small learning rate results in slow convergence while a too big one can result in oscillations and make us avoid convergence.

Activation functions

The choice of the activation function is very important as it can avoid some issues during training. In this thesis, we chose to rely on the ReLU activation function for its properties against vanishing and exploding gradients [19], [20] and the added bonus of coupling well with potential Kaiming weights initialisation. [21].

We will however use the softmax activation function on the output layer. Indeed, the output we want is a probability distribution over the two possible classes and softmax will map the non-normalized outputs of the two output neurons of the output layer into a probability distribution.

Additional general tools

We will here make a quick reminder of some basic tools used currently in Deep Learning to increase performance and/or fight common encountered problems in training.

- Adam : While the basic optimization algorithm of Neural Networks is the gradient descent we evoked earlier, more sophisticated algorithms exist that can improve training. We here decided to use the Adam optimizer [22]. Adam keeps a different learning rate for each weight and biases and adapt it as the training goes.
- Batch Normalization : modifies the outputs of each layers by normalizing them, adds stability to the learning and can help fight exploding and vanishing gradients. [23]
- Dropout : This technique consists in dropping random neurons of the network at each training step. This forces the network to be robust and not rely on a few specific connections, which helps with overfitting and generalization in general.

3.2.3 Convolutional Neural Networks

The neural networks we described earlier are called fully connected neural networks. They are considered deep if there are many layers between the input and the output ones. While fully connected

layers are performant, they do not always take advantage of the structure of the data presented to them. Convolutional layers are designed to do so. They have notably been ground breaking in computer vision tasks, but can be used on 1D data as well. Their inclusion in common fully connected networks yields what is called a CNN : a convolutional neural network.

Convolutional layer

Handling data such as images and sounds as normal inputs with NN would quickly be untractable. For this reason, convolutional layers focus on regions of the data and extract from them some information. The same linear transformation is applied locally everywhere while preserving the signal structure. These linear transformations are typically convolutions.

For a 1 dimensional signal like ours, given an input vector $\mathbf{x} \in \mathcal{R}^W$ and a convolutional kernel $\mathbf{u} \in \mathcal{R}^w$, the discrete convolution $\mathbf{x} \circledast \mathbf{u}$ is a vector of size $W - w + 1$ such that [24]

$$(\mathbf{x} \circledast \mathbf{u})[i] = \sum_{m=0}^{w-1} x_{m+i} u_m$$

These convolutions can capture local features with limited added cost for inputs with many variables. Common practice is to stack these convolutional layers on top of each other to extract low level features first, then from these low level features extract more high level ones that can then be fed to fully connected, dense layers.

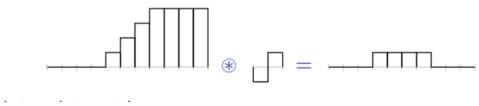

Figure 3.5: 1D convolutional operations

Pooling layer

Pooling layers are another way to reduce the complexity of outputs of layers. They also counter one of the issues of convolutional layers, which is the way they rely on precise positions of the inputs variable, making them more vulnerable to small changes in positions. Pooling layers take a kernel of a given size and outputs a single variable for this ensemble of features, based on a simple operation, such as the maximum value or the average value of the inputs covered by the kernel. This brings invariability of location to the features output by the association of a convolutional and a pooling layer, which helps with generalization.

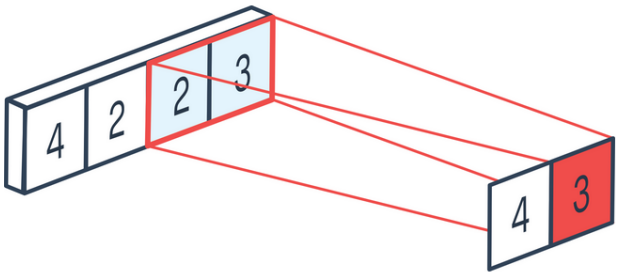

Figure 3.6: 1D max pooling operation

3.3 Performance metrics

Performance metrics are an essential part of model building. They allow to compare models to one another to determine which is best, and they give insights on the advantages and shortcomings of the model. But choosing the correct one can be tricky, as just one metric can lead to a biased perception of the performances achieved. In our case, for example, our dataset is made of 97% of sleep samples, while only 3% of the samples represent the positive class to detect. Which means that a really bad classifier predicting each sample as sleep would get by default a 97% accuracy score, while being terrible at its job.

In this section we will define classic metrics to use in binary classifications cases . We will highlight their shortcomings with regard to our specific task and come up with a solution to counter these shortcomings. We will also discuss the disadvantages brought by this solution and conclude on the way to consider our results all things considered.

3.3.1 Basic performance measures

In binary classifications, it is common to classify the predictions in four different categories, as illustrated in Figure 3.7

- True Positive (TP) : elements predicted as positive that were indeed positive
- False Positive (FP) : elements predicted as positive but that were negative
- True Negative (TN) : elements predicted as negative and that were indeed negative
- False Negative (FN) : elements predicted as negative but that were positive

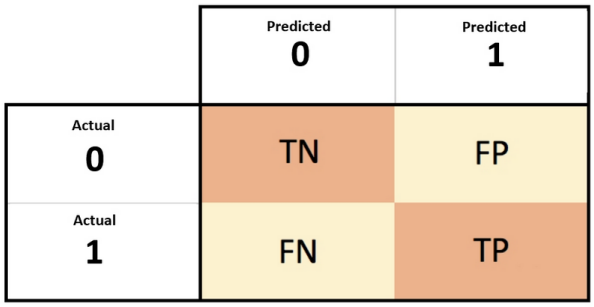

Figure 3.7: Confusion matrix

Based on these definitions, we can define a few commonly used metrics that give several insight on the performances of a classifier.

The first one is the accuracy, which is simply the amount of correct predictions over the total amount of predictions :

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

As we explained earlier, this is not a good measure of performance in our case, because of the strong imbalance there exists in our dataset distribution over the two classes.

For this reason, we will use the two following metrics for our experiments :

$$precision = TP/(TP + FP)$$

and

$$recall = TP/(TP + FN)$$

In other words, the precision measures how reliable a positive prediction is, while the recall measures the ratio of positive elements identified as such by the model.

These two metrics give us enough information about the performances of a model to be able to understand how it would behave in real life. However, there is a tradeoff between the two of them that can lead to difficult choices when looking at two similar models. For example, choosing which model is best between a model A with a recall of 0.7 and a precision of 0.5 and a model B with a recall of 0.6 and a precision of 0.6 is not trivial. This is a problem because in our case, precision and recall are both equally important : making too many incorrect positive predictions could lead to a waste of time for both the patient and the doctor by identifying a healthy patient as ill, while not identifying enough micro-awakening could lead to ill patients not getting the right treatments. For this reason, we chose to use the F1_score to combine both metrics into one for model comparisons purposes. The F1_score is an harmonic mean of the precision and recall :

$$f1\_score = \frac{2 \times precision \times recall}{precision + recall}$$

We can therefore use the f1_score to compare models to one another on the basis of the recall and precision metrics.

3.3.2 Precision Recall curve and AUC

Random Forests and Neural Network do no output directly a class. Instead, Random Forest output a number between 0 and 1, representing the probability for the input to be positive. Similarly, the neural network, after use of the softmax activation function, outputs two probabilities, one for the sample to be each class, with the sum of these 2 probabilities of course adding to 1. By default, the prediction is considered to be positive if the output of the Random Forest is above 50%. In the case of the Neural Networks that have two outputs, one for each class's probability, the outputed class is considered to be positive if if the output corresponding to the positive class is higher than 50% by default. While these are the default settings, this threshold can be changed.

Now changing this threshold does mean that the performances of the model will change as well. Indeed, setting a lower threshold means that more elements will be classified as positives. While this will probably increase recall, it also means that more False Positives will be predicted, hurting the precision. We therefore can plot what we call a Precision Recall curve (PR curve in short). This curve plots the precision to the recall for various values of thresholds. An example is given on Figure 3.8.

This brings an additional way to evaluate models : instead of judging them on a single instance of their recall/precision performances, one can measure the area under the curve of the PR curve. This shows how well the model performs in general for various thresholds. An ideal model would classify all positives as 1 with output probabilities of 100% while negatives would all be predicted as such with an output of 0%. Therefore, the curve would actually follow the upper and right sides

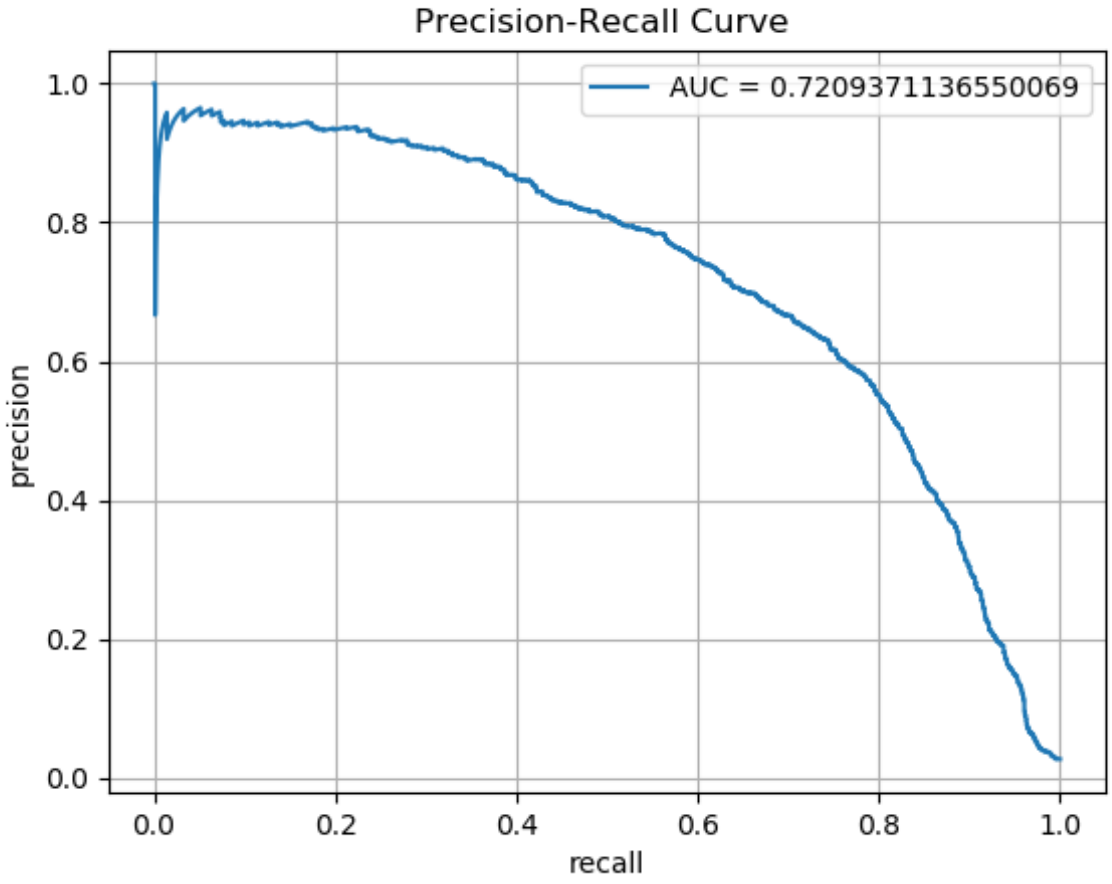

Figure 3.8: An example of Precision Recall curve

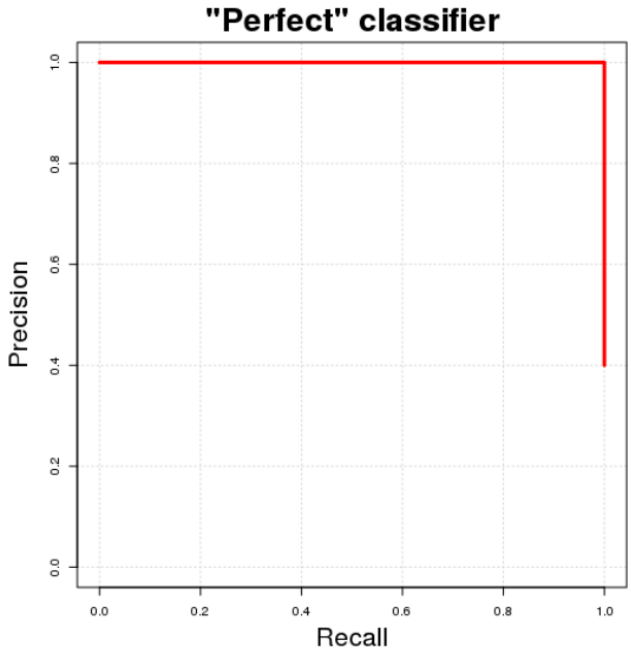

Figure 3.9: A perfect classifier has an area under its Precision Recall curve of 1 as the recall and the precision are always worth 1

of the graph, making the ideal model's PR curve cover 100% of the surface available, as shown in Figure 3.9.

Similarly, the worst model would classify all samples perfectly wrong with a 100% certainty on them, making the corresponding curve hug the left and lower side of the graph, for an area under curve (AUC) of 0%. Following this idea, the better a model is, the higher its AUC to the total surface ratio is. This allows to compare models in a more general way.

3.3.3 Problems encountered and our proposed solutions

The problem with the standard way to compute Precision Recall

When we got our first results from Random Forests, they were pretty bad. By plotting the PR curve of the model, we realised that the problem came from the default 50% threshold that was way too high. Lowering it from 50% to about 17% led to drastic improvements over recall and precision.

We then analysed the results of these predictions and visualized them in Apios. We observed that while the model did make some legitimate errors, a lot of them were actually related to some offset between the zone where an event was predicted on the Jawac signal, and the zone labeled as positive by the doctor, making predictions that are right next to a positive region, or even overlapping on it, counted as errors. (see Figures 3.10 , 3.11 and 3.12, for examples). This was hurting the performances for the wrong reasons, making comparison between models less interpretable, and overall didn't make a lot of sense given the primary goal of the project. We therefore decided to take some inspiration of the way performances are evaluated in object detection in image.

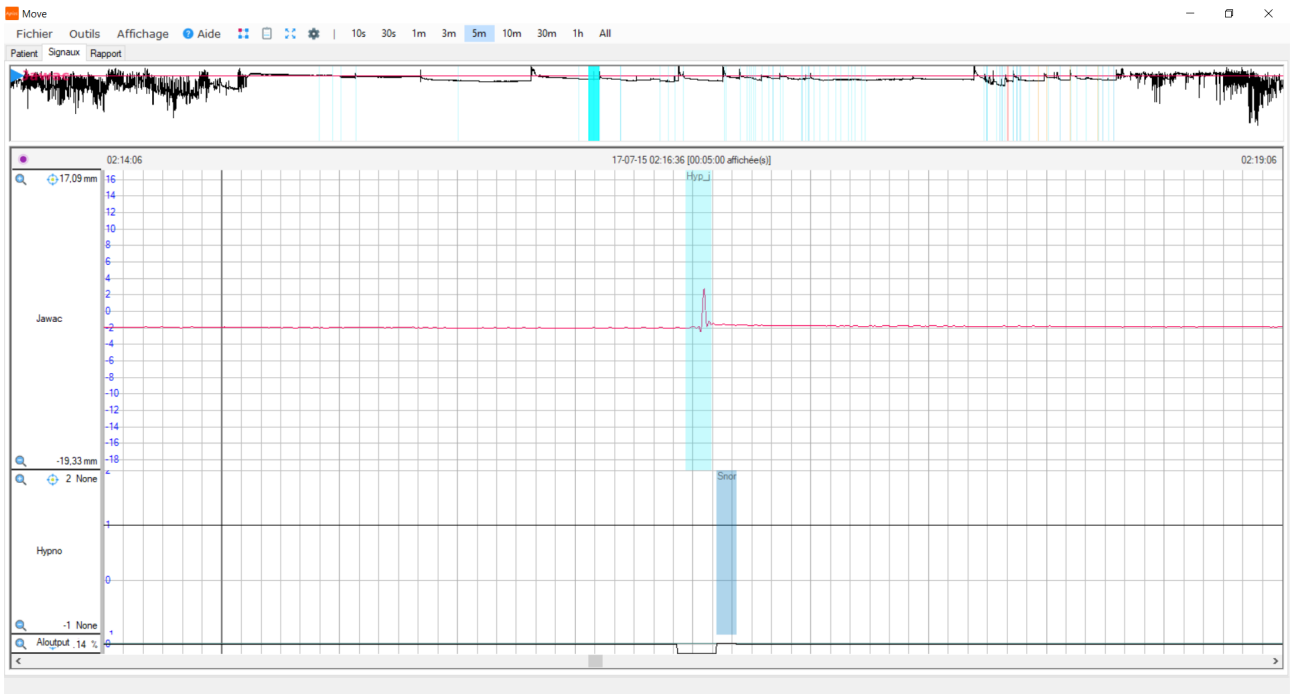

Figure 3.10: A prediction (dark blue highlight marked in the hypno signal) misses the arousal location (light blue highlight on the Jawac signal) by very little and will be counted as an error

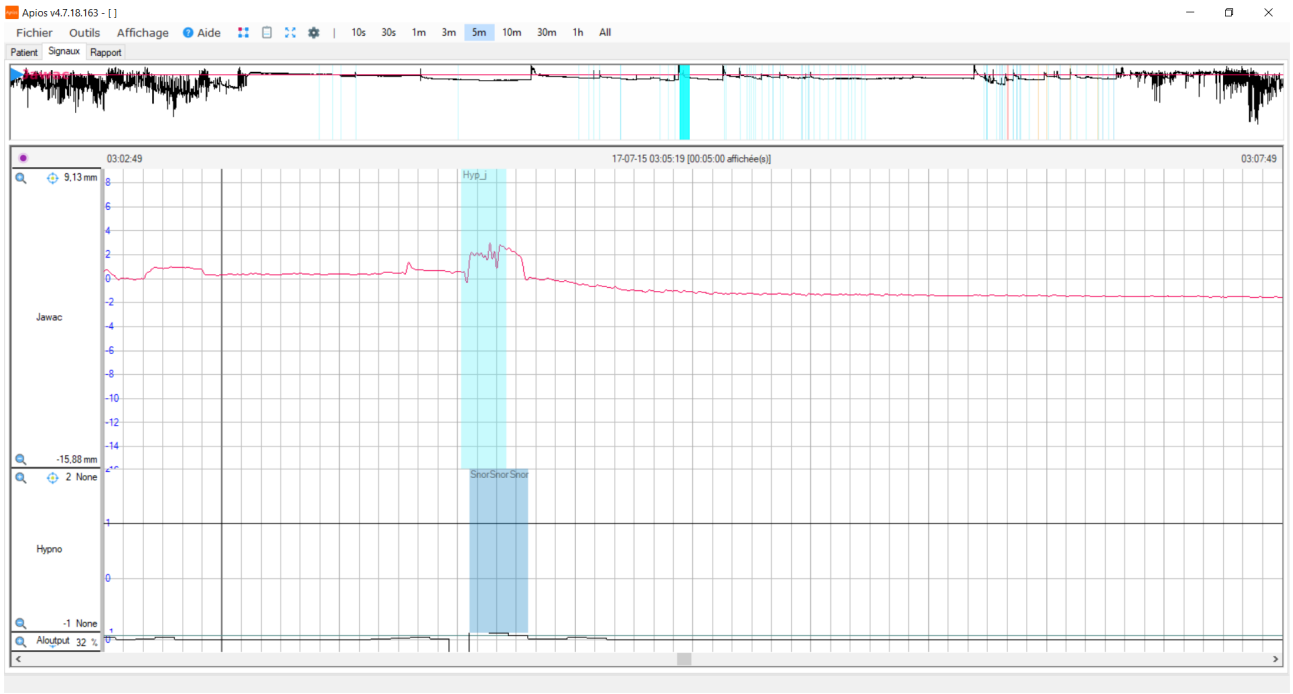

Figure 3.11: The predictions (dark blue highlights marked in the hypno signal) overlap the arousal location (light blue highlight on the Jawac signal). The 3 samples predicted as arousals will be counted as 2 true positives and one false positive, as the right most sample does not correspond to a positive sample. Furthermore, the left most sample of the true positive samples is predicted as negative and will therefore count as an error as well. All in all, 2 mistakes and 2 good predictions are counted in this image.

Jaccard coefficients

In object detection in image problems, a square zone is predicted and is expected to be at the exact same location as the ground truth represented by a square delimiting some object the model should find in the image. If the predicted regions do not perfectly overlap, the loss is computed as the "Intersection Over Union", meaning that the ratio between the intersection of both regions and their union is computed (see Figure 3.13). This coefficient is also called the Jaccard coefficient. If this coefficient is above a given threshold, the prediction is considered a success (a TP), and a failure (a FN or a FP) if not. We decided to use the same idea on our problem. We considered adjacent positive labels as a single unified positive region, and applied the same treatment to predictions. We therefore switched our view, performance wise, from a serie of 5 second samples to a serie of zones of various lengths. We converted these in two binary vectors, one of predictions, one of ground truth labels. From these we were able to compare each predicted positive zone to the ground truth vector and see if the predicted region overlapped any ground truth positive region. If yes, then the Jaccard coefficient was computed, and if it was above a Jaccard threshold, then the whole predicted zone was considered as a single true positive. Otherwise, it was considered a false positive. We then applied the same reasonning the other way around to detect false negatives. Figure 3.14 illustrates False Negatives, False Positives and True Positives cases. One can notice that on this system, counting True Negative doesn't really make a lot of sense as these regions can be quite long, as opposed to the short micro awakenings.

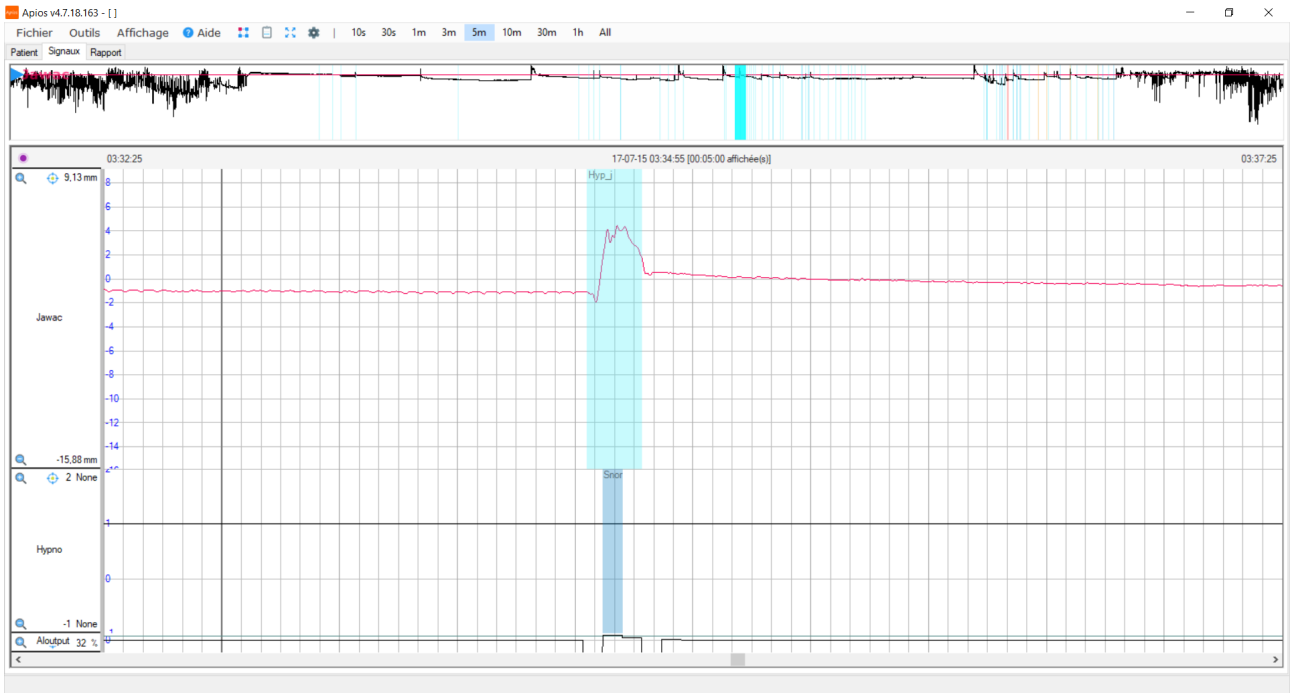

Figure 3.12: A prediction (dark blue highlight marked in the hypno signal) accurately predicts the center of the arousal location (light blue highlight on the Jawac signal) but fails to predict the 2 positive samples to the left and right of the correct prediction. In total, 2 errors and 1 correct prediction are counted in this image.

Figure 3.13: Jaccard coefficient core idea in object detection in 2D images

Jaccard Precision Recall Curves

We were therefore able to compute the performances of our models in a way that was more fitted to the nature of the problem, where a slight temporal distance between the actual positive zone and the predictions does not matter too much as long as the zone has been detected.
From there, we were able to create our own PR curve plots, by varying the threshold on probability zones, only with recall and precision. While these provided valuable visual informations on how models performed, they were not problem free. Indeed, this new way of computing the classic con-

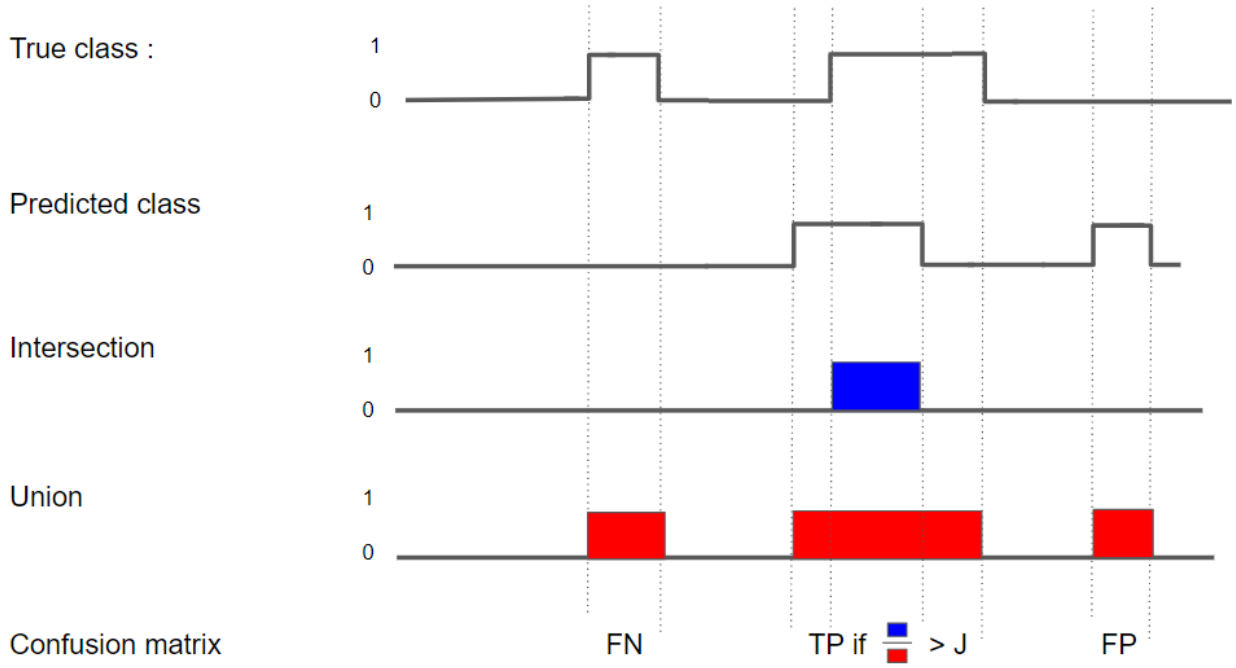

Figure 3.14: Our implementation of a 1D Jaccard system. A zone is considered a true positive if the intersection (the blue surface) over the union (the red surface) is bigger than a given Jaccard coefficient J. Adjacent predictions are grouped together to form positive regions rather than individual samples and count together as one error or one success.

fusion matrix elements led to some counter intuitive behaviors for the curves. When the threshold for predictions lowers, more and more elements are seen as positives. This mean that the amount of regions seen as positive in the prediction vector will rise as well. At first, this improves recall and hurts precision, as usual. But in this Intersection Over Union context, we sometimes have positive predictions zones that suddenly joint as the in between region suddenly becomes positive as well. This creates sudden large positive predictions zones, while the potential ground truth positive region in the ground truth vector doesn't grow. This makes the denominator of the Jaccard index (the union) grow, while the nominator (the intersection) doesn't. As a result, lowering the prediction trigger can make some predictions go from a Jaccard index big enough to be counted as a TP, to a Jaccard index too small and effectively make some predictions previously seen as correct become incorrect in that regard. This translates in the recall going down as the prediction threshold becomes too small, and we can therefore observe the phenomena of Figure 3.15, where the PR curve behaves normally before making a turn around.

To adress this, we decided to simply not consider the curves after signs of taking a backward trajectory. We however had other problems with these curves : they notably wouldn't necessarily stop far enough in the graph; or start early enough, for their AUC to be of any use.Indeed, the curves not starting/ending at one of the sides of the graph means that when computing the AUC we lose a whole portion of the graph, making the resulting AUC worse than other models of similar qualities. We therefore decided to use the PR curves of the Jaccard coefficient based confusion matrix to strictly visually compare models, as well as to evaluate the performances of such models on the different patients composing our testing set. This will be notably useful to identify recordings that were particularly poorly evaluated in order to analyse the errors made and try to find a pattern.

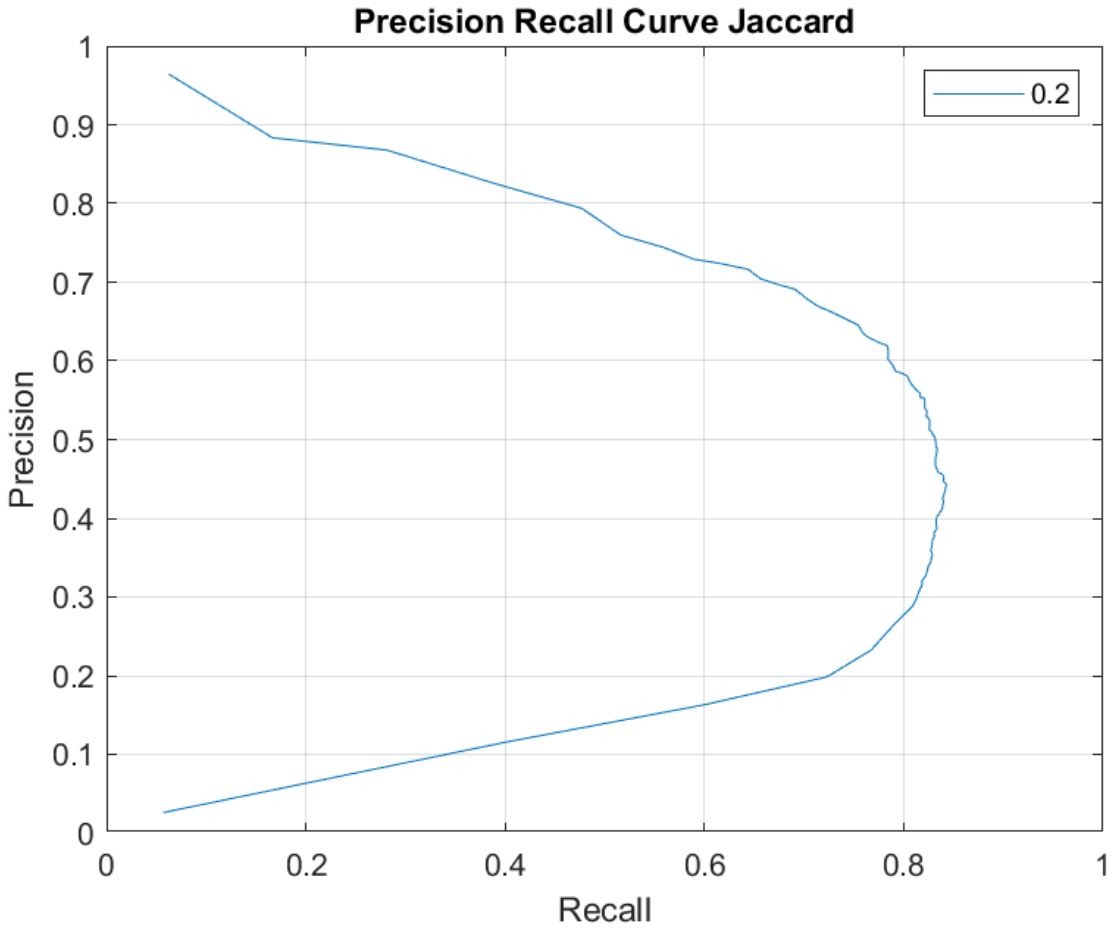

Figure 3.15: Precision Recall curve using Jaccard coefficient of 0.2. The constant augmentation of samples predicted as positives leads to the Jaccard index denominator to grow because of the aggregation of adjacent predictions in a single positive predicted zone, which at some point lead to the recall to drop as the Jaccard index falls under the threshold needed.

These results will be detailed in the corresponding section.

3.3.4 Conclusions

In order to avoid the problem of acceptable predictions being counted as errors, we will use the Jaccard coefficient method to consider what is a TP, a FP and a FN. The model comparisons will be made through the use of PR curves in both systems, with the Jaccard PR curves being used for additional visual information. A model's best performances will be estimated by computing the best recall-precision pair the model can offer. This pair is identified through the use of the f1_score, to provide an objective and systematic way to define what is "best". These precision and recall metrics will be computed in both the classic and the Jaccard system, with the Jaccard system being the one that matters the most to us here.

Chapter 4

Previous work and state of the art

In this section we will display a literature review of the state of the art methods that have been used in similar contexts. We will also review the current algorithm implemented and used by Nomics and its shortcomings, motivating the new approach.

4.1 Current algorithm and limitations

In 2008, F.Senny finished the PhD thesis [1] that was concerned about the study of the Jawac signal and its then potential application for various sleep diagnosis, including the diagnosis of SDB and the detection of arousals. This part of the study was motivated by the observation that the salient jaw movements, the sharp or smooth closure movement(s) of the mandible translating arousals, are often associated with SDBs. The idea was to detect these signs of arousals (Figure 4.1) and treat them as indicators of a suspicious area prior to that arousal. Looking at what happened before these signs would help focus on relevant parts of the recording and identify SDBs more easily. It is in this context that Senny decided to develop an algorithm to detect arousals.

The algorithm implemented by Dr Senny was based on the tracking of movements of significant amplitude. For this, he used the wavelet transform [25], a classic tool in the analysis of biosignals [26]. Senny used a first derivative of the continuous wavelet transform to detect movements of the jaw in general. This wavelet has the two advantages of removing the mean component and to associate a curve with each edge of the signal (the opening and closing jaw movements.). This curve is a chain of the maximas of the wavelets of the signal. This allows to identify jaw movements. The area under the aforementionned curve later allows to classify these movements. Indeed, areas under sharp or big amplitude movements are bigger than areas under smooth and low amplitude movements. A representation of this is shown in Figure 4.2

Using this method, Senny obtained on 34 recordings a recall of 80% and a precision of 75%. It is here important to note that Doctor Senny worked on a different dataset than ours, dataset that got lost in between his work and ours. This dataset included a much higher number of recordings. It is also noteworthy that this analysis revolves around the computation of two parameters : the amplitude of the jaw movement and the increase of this movement compared to moments before. These parameters were chosen on a rational basis but could be potentially complemented by additional ones.

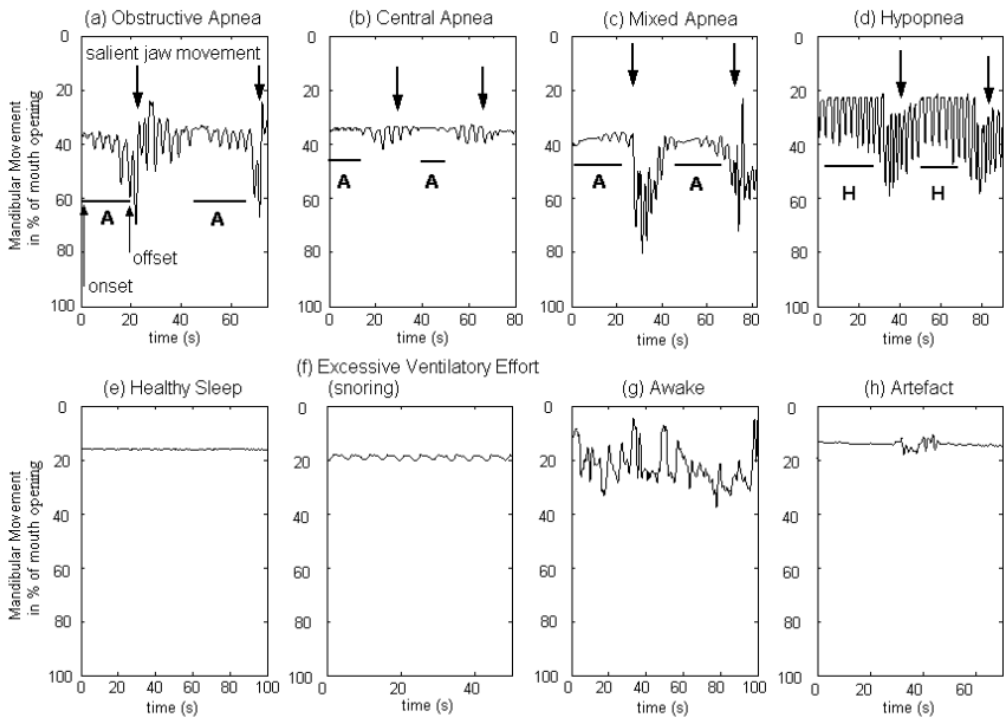

Figure 4.1: Typical sleep apnea and hypopnea (SAH) patterns on Jawac: obstructive (a), central (b), mixed (c) apneas and hypopneas (d). The salient jaw move- ment are pointed by an arrow. Typical not-SAH patterns : (e) healthy sleep, (f) snoring, (g) awake, (h) artifact (a swallowing).

4.2 Related work and state of the art results

The use of Deep learning in the classification of bio-signals is not a novel idea [27] [28]. The improvements in machine learning and the rise in available datasets of signals like EEG, or ECG(electrocardiograms) triggered a flow of studies on the classification and identification of various pathologies, or other tasks [29].

On the subject of sleep apnea specifically, various approaches have been explored, but none on a signal similar to the one developped by Nomics. Moreover, the subject of the papers we found was mostly about the detection of SDB directly, and less often with the detection of the micro awakening themselves [30]. It is also worth noting that according to [27], sleep scoring, while having been explored, hasn't been studied enough for a typical best performing architecture to stand out.

Three main signal approaches stand out : The study of the signals composing the PSG, the study of the EEG, and the study of the ECG (electrocardiogram) [31]. Applying varied deep learning techniques onto one or several of these bio-metrics have been the most common approach in recent literature, to the best of our knowledge. It would be tempting to divide the literature review on these different bio-metrics, but we have to remember that we are here tasked with working on a specific signal. The concern of which signal is more suited for the detection of arousals is therefore irrelevant to us. What is, however, interesting to us, is the way the data was handled and the architectures used to reach the results. For this reason, we decide to go through this review with a different angle. We will be reviewing first approaches based on a pre-processing of the bio-metric used, would it be

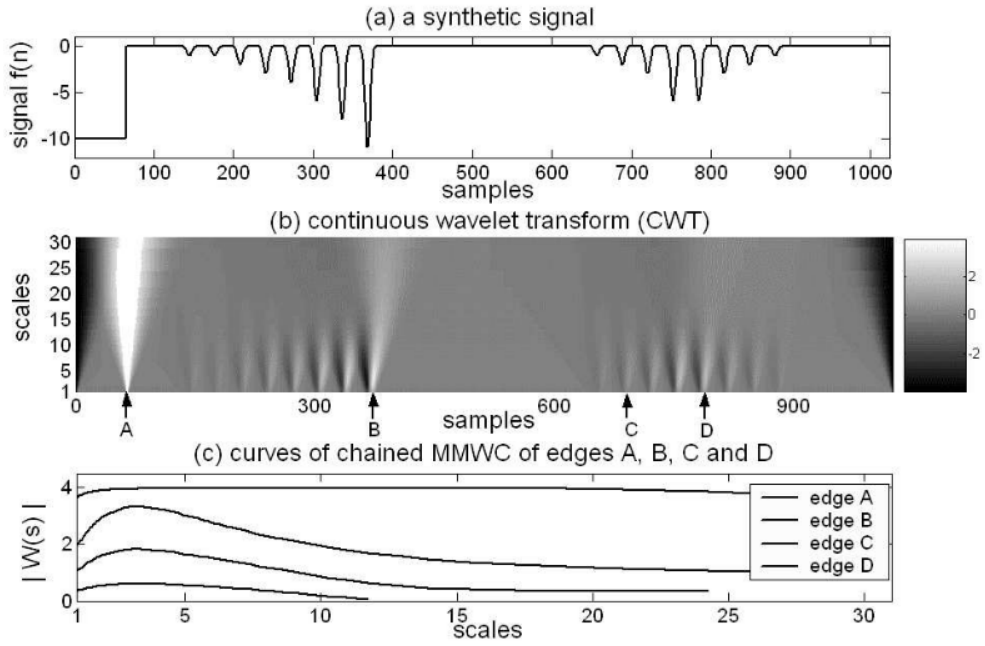

Figure 4.2: [1] (a) A synthetic signal, (b) its continuous wavelet transform (CWT) and (c) the chains of the wavelet transform modulus maxima in function of scales of four edges noted A, B, C and D. Salient jaw move- ments (edges A, B and D) and respiratory effort (edge C).

PSG's signals, EEG or others, and then we will review a few papers that tackled the problem with a raw signal. For all, we will summarize the type of model used and the results obtained.

4.2.1 Approaches based on preprocessed signals

In [32], a RCNN leverages the use of several channels of the EEG. The paper describe the use of both a raw waveform and a spectrogram representation of the data as inputs for the detection of apnea events in general, in order to determine the AHI (Apnea Hypopnea Index [33], a measure of the frequency of occurence of apneas and hypopneas in a night of sleep). This RCNN managed to predict 1 second long windows on the basis of 30 second longs inputs with a 81.9% total accuracy.

In [34], a tracheal sound is used for the detection of apneas. It uses labelled PSG for scoring 1 minute long segments that are then turned into spectrograms before being fed into a 2D CNN. This method showed a 95% agreement with manually scored PSGs for the computation of the AHI.

4.2.2 Approaches based on raw signal

In [31],the ECG was in a first attempt extracted from the PSG and segmented in 10 second long segments with no overlap. The ECG signals were converted into 2D spectrogram images to generate 2D input signals using the following short-time Fourier transformation:

$$x[n,k] = \sum_{m=0}^{l-1} w[m] \cdot x[n+m] \cdot e^{-jm(2\pi k/N)}$$

Various deep learning architectures were then applied on this input, among which 2D CNN, GRU, LSTMs. Their best architecture however ended up being a 1D CNN which was trained on the raw ECG data which had just undergone a batch normalization. This ended up being the single best model of the whole experiment, reaching a 96.3% accuracy on apneas and hypopneas. They conclude the experiment on the following note :" We can recommend 1D CNN [...] for automatic detection based on the time series signals such as ECG and other physiological signals. ".

In [35], the same kind of approach is made. The ECG is extracted from the PSG and a CNN is applied to the detection of obstructive apneas. Here the data is segmented in 1 minute long segments with binary labels indicating whether or not the segment encompasses an OA or not. They obtain a final precision of 99% and a recall of 97.8%

In [36], signals are extracted from a PSG then filtered. They are later fed into a LSTM network. The data is segmented into 30 seconds long segments, which gets its labels from a skilled technician. If the end of that window has been labelled as apnea, the whole window is labelled as such. They manage a total accuracy of 99%.

4.3 Motivation for a new approach and potential restrictions

While the results presented in the section above show excellent results, they still for the majority depend on PSG metrics, and therefore suffer from the major problems of the PSG : the limited number of bed in specialized sleep clinics, the invasive sensors and the overall complexity of the process for the patients and doctors. Nomics on the other hand tries to create a device that a patient could take home, wear during the night and bring back for automatic analysis, skipping at least in a first time the PSG diagnosis.
These results also come from models that were trained on the same signals that were used to label the datasets used, while we want to train a model to imitate a human diagnosis made on a EEG by only using the Jawac signal.

Furthermore, the vast majority of the work done on the subject focuses on the detection of SDB, and not on the detection of arousals. But most of the time, the final goal is to identify suffering patient through the use of the AHI, which while often correlated to the micro awakenings is not the same thing.
The good news we take from these related works is that it seems definitely possible to apply Deep Learning algorithms to the detection of SDBs, which hints that the detection of micro awakenings should also be possible on the Jawac given the relation between the two and the previous work done by Senny. We have an example of a success story close to our case in the shape of the work of Julien Simar [10] : in his Master Thesis, he used the Jawac signal to classify portions of signal as sleep and awake phases, continuing another aspect of the thesis of Senny.

While the algorithm of Dr Senny works rather well, it is based on only two pre computed features. This means we reintroduce a systematic aspect on the definition of what is obstructive or not, defined by thresholds on respiratory effort, which comes in the way of the interpretation of the results. The strength of Deep Learning is to automatically extract features and combine them in the most efficient way to maximize the quality of the output. This combined to the success of these methods in analog cases in both Jawac and classic bio signals provides incentive to try out these modern methods for better classifications.

Chapter 5

Data processing

The preprocessing of the data was done in Matlab 2020a. This choice was motivated by the various tools already implemented by Nomics in Matlab. The data is composed of 95 Jawac recordings of various patients that registered a night of sleep.

The main steps we followed were the following : from the files containing the recordings of the patients nights, we extracted the Jawac signals and their corresponding hypnograms. We also gathered the files that held the actual events we were interested in for each of these recordings. After cleaning, pre processing the Jawac signal, and delimitating the zones we wanted to work with, we extracted the events we wanted to predict from their mk3 files and created a new signal : the markings signal. This signal has the same dimensions as the Jawac signal and classifies every time step of the Jawac signal. It will therefore be our basis for determining our Y_true values, while the Jawac signal will be used to determine our X_values.

Each of these steps will be explained in details in the following sections.

5.1 Jawac Signal

Nomics stores its Jawac signals recordings in EDF files. The EDF format is the European Data Format and is a standard of informatic files made for the storage of medical and biological data organised as temporal series, such as physiological signals. Nomics uses one EDF file for each night of recording made on one patient.
The EDF files we were handed over by Nomics belonged to a larger PSG analysis, annoted by a doctor. While this was a good basis for a dataset, it needed some various pre processing steps. Luckily, the work done by Julien Simar [10] on the detection of sleep and awake periods on the same data needed the exact same kind of preliminary work. We therefore were able to reuse most of his work on the Jawac signal treatment. We will here make a brief summary of the modifications made to the Jawac signal we ended up using.

5.1.1 Extraction of the Jawac signal

The EDF files contained much more than the Jawac signal we were interested in. In fact, no less than 46 signals sampled at 500Hz were present in the original files. This made them too big to handle and open with Apios and Matlab. The EDF files were therefore cleaned from the additional signals that were of no use.

Downsampling of the obtained signal

The original Jawac signal in the EDF was sampled at 500Hz, while Nomics tool samples it at 10Hz. Since the goal of the work is to have Nomics being able to use the models on these output signals, the signal had to be downsampled.

Extraction of Hypnogram and synchronization

Each EDF record is associated with a hypnogram file. These files record the sleep stages of the patient at any time of the night. They can therefore easily be turned into a 1 dimension vector of binary values of the same dimensions as the Jawac signal. This is what Simar did in his work. He also had to synchronize both signals as they did not necessarily start or end at the same time. Therefore, only the overlapping zones between the two signals were kept. The time stamps referencing time steps were recomputed so that the common reference would be the start of the day starting the recording. This allowed for easier comparisons. The hypno signal was finally added to the EDF files alongside the Jawac ones, so that visualisation of both was possible simultaneously. This is important for us as well as we are not concerned one bit with awake zones and need to ignore them in our training and evaluations processes.

5.1.2 Jawac signal preprocessing

Finally, some extra processing was done on the Jawac signal values :

- Calibration was applied so that the signals could be compared on equal ground. This is essential, as supervised learning algorithm will need the same value to represent the same thing from one recording to another for a given feature.
- The values our Matlab code handles are digital values resulting from the conversion of the physical values recorded by the sensor by an analog to digital converter. Nomics wanted to be able to analyse the decisions made by the models we would find and understand the process. This required the data to be turned back to their physical values so that the interpretation would be made easier for later works.
- Finally, the data was normalized as it has been shown to drastically improve the learning process of Deep Learning models [37]

5.2 Dataset Labeling

From the simultaneous signals registered in the original EDF, a specialist later conducted an analysis in order to label specific time zones which, according to his expertise, corresponded to various types of sleep events. These labelled periods were referenced in two text files named PNEUMO and NEURO files, one of each for each EDF file. From there, we needed to extract the labels from their respective files and transform them into a *Y_true* signal parallel to the Jawac signal and representing the corresponding classes. This signal will then follow some transformation before being finally used jointly to the Jawac signal to build the dataset.

5.2.1 Marking of outliers

Sometimes, a part of a recording is not recorded correctly. This can happen when the sensor get disconnected during the night, or falls off before being put back on, and so on. These accidents create periods of recording that are not usable for any kind of analysis and that should be left out. For this purpose, there exist a marking label in the mk3 file format that defines "Out of Range" regions. The Figure 5.1 shows such a case.

From the previous work of Julien Simar we inherited mk3 files associated with the same recordings we are working on. In these files, some Out of Range regions had already been indicated. We used these same mk3 files to benefit from this pre processing.

We will consider "Out of Range" marked regions as a specific class in our end *Y_true* signal, so that we can later ignore these regions in our learning process.

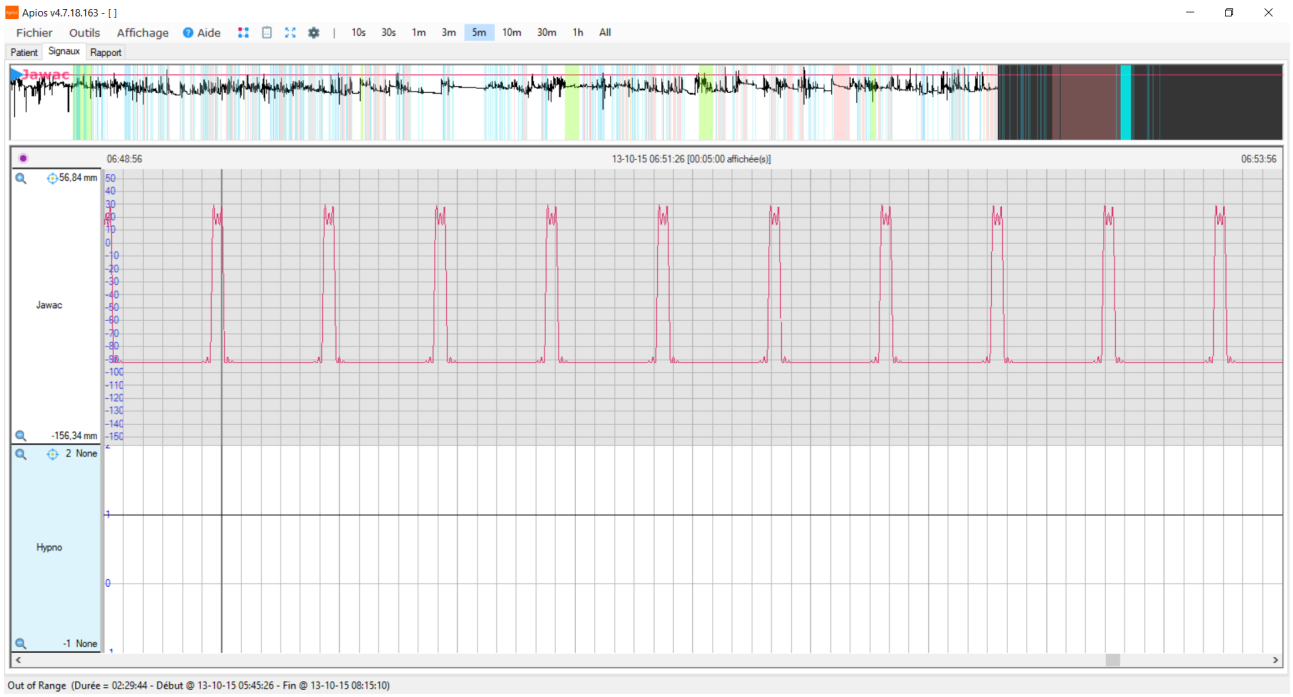

Figure 5.1: Faulty sensor output results in the zone being marked as OOR ("Out of Range"). This is represented in Apios by a grey highlight.

5.2.2 Extraction of the Pneumo and Neuro files markings

The mk3 file format allows a referencement of events associated with a label that can be read by Nomic's software, Apios. In this format, each event is associated with two timestamps, one for the beginning and one for the end of the event. The events are also associated with a label, a signal and a date. Apios uses these files in conjunction with their matching EDF files to highlight on the visualized Jawac signal the time periods where something happens with the matching label.

We extracted the events referenced in the PNEUMO and NEURO files and converted them to the mk3 format, making them visible in Apios.

Each different event was attributed a different marking label so that they could all be identified uniquely.

5.2.3 Dubious markings

In some cases, the markings made by the doctor can seem to be out of place to an experimented eye. This can come from the doctor using thresholds for his decision making and having events barely below these thresholds being ignored when they shouldn't, it can come from human errors, and so on.

These are different from "Out of Range" zones of the signal as the signal itself is not in doubt here : we can use this piece of data as context. But using them as a basis for determining the class of a given signal segment might be ill advised. To avoid using these chunk of data at inopportune times, some pieces of the signal were marked as " Dubious marking".

This is another heirloom we got from Julien Simar's previous work. Indeed Nomics had already looked over our recording and has identified several of these dubious markings.

We therefore also reused the "Dubious markings" that were present in the original mk3 files left by Simar's work. An example of this is shown in the Figure 5.2

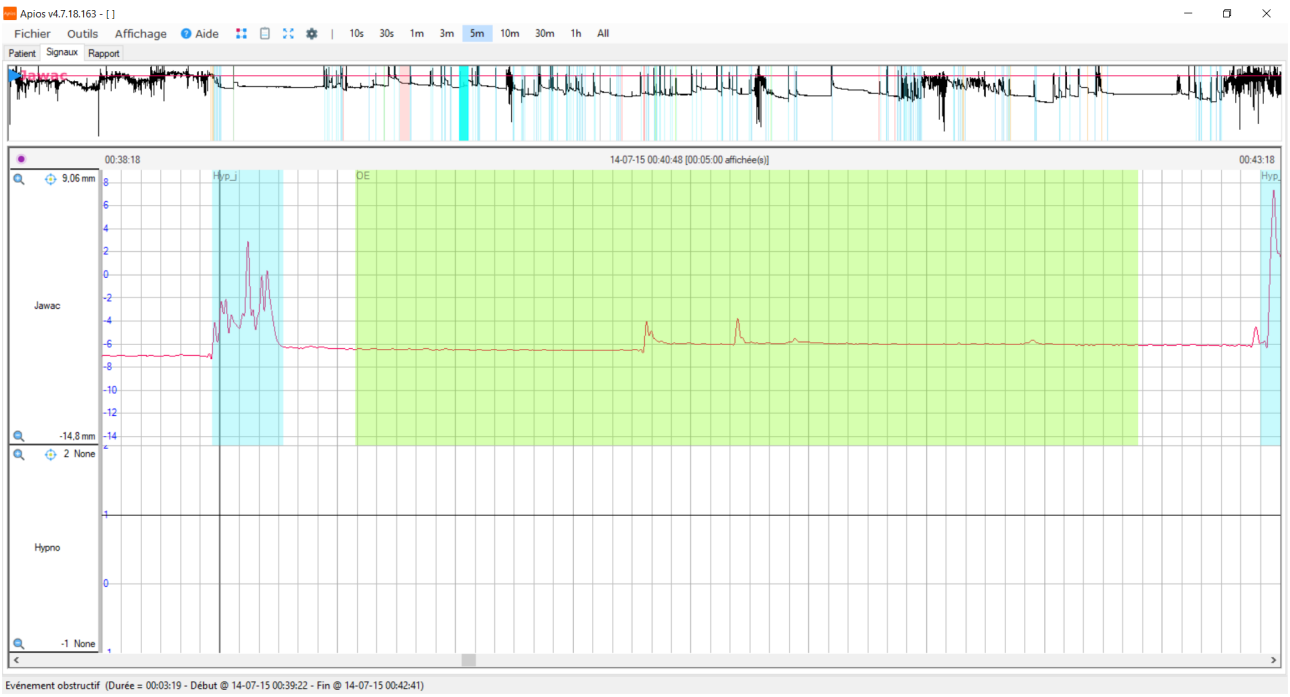

Figure 5.2: A part of the signal marked as dubious by Nomics (light green)

In addition, since the work that interests us here doesn't concern the parts of the signals where the subjects are awake, we decided to mark all awakened zones as being "Dubious". This is done thanks to the hypno signal, that we added earlier in the EDF, and that translates the state of awakening of the patient at any given time. We extracted this signal and used pre-built functions to mark the awakened zones in our markings signal.

5.2.4 From mk3 markings to the marking signal

After all these considerations, we ended up with each EDF file matched with a modified mk3 file containing markings for each individual delimited signal portion that stands out.
These markings were then transformed into a signal of the same length as the Jawac signal, so that for each time step, we would have a corresponding label. This signal is saved in its corresponding EDF file, so that the final versions of both the Jawac signal and its labelling can be consulted easily and visualized in Apios.

After all these steps, we end up with a base data made of several EDF files, each corresponding to a patient night of sleep recorded, and holding two signals : a processed Jawac signal and a parallel signal representing the current class the Jawac signal is associated with for each time step. These two signals are now ready to be transformed in our X and Y_true values to constitute workable datasets. The Figure 5.3 displays the two signals side by side. The light blue highlights on the Jawac signals are the visual representations of the events referenced in the mk3 file associated with the Jawac. These represent micro-awakenings.

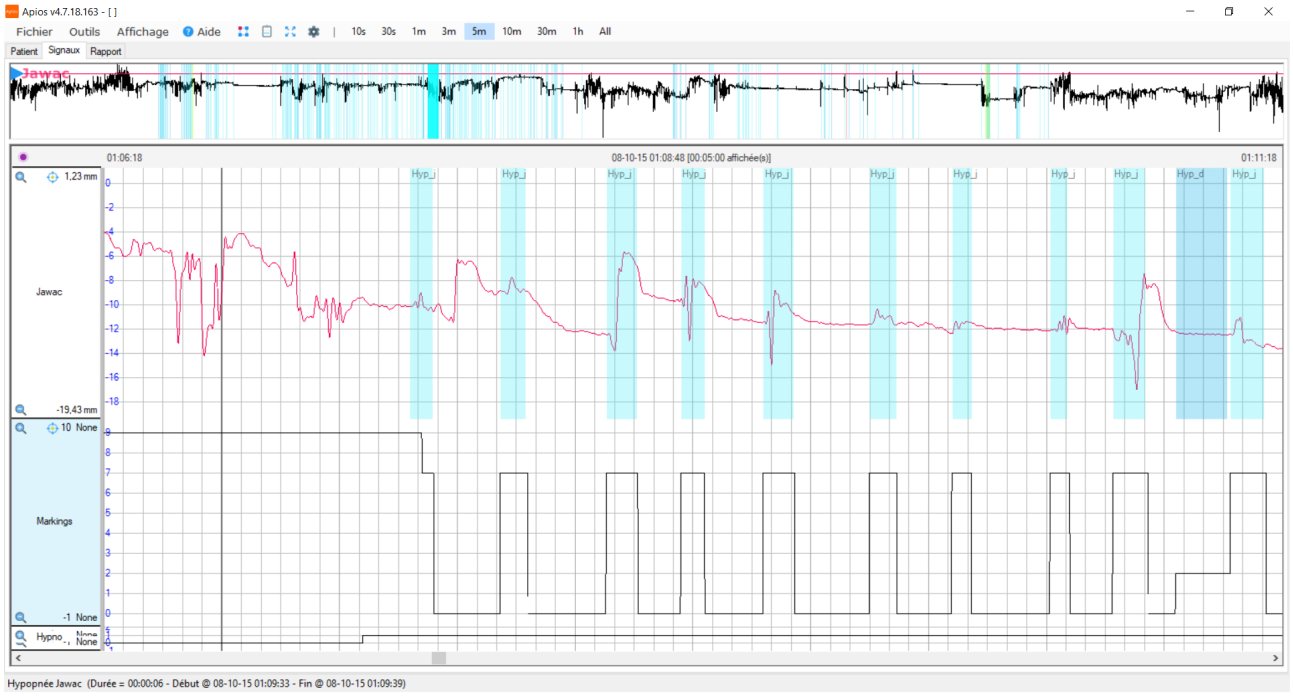

Figure 5.3: The Jawac signal and its associated markings signal

5.3 Creation of the datasets

For this work we used the *sklearn* library for the common evaluation metrics as well as the Random Forests, while we used Pytorch for all the Deep Learning related work. Since the data needs to be presented in the same format for both frameworks for training purposes, we can generate just one dataset for both uses. This dataset must present the entry data in the format $(N\_Samples, N\_features)$, where $N\_Samples$ is the amount of samples in the dataset and $N\_features$ is the amount of features in each sample.

5.3.1 General aspects

We quickly decided that the central system to build the dataset would be a sliding window of a given size going through each vector of marking signal, our Y values. For each step, if the sample delimited by the sliding window was deemed usable, we would transform the matching Jawac sample into a X input variable and extract the Y value before concatenating both into their respective X and Y_true vectors. But some central points still needed to be addressed.

The first question we had to ask ourselves was about the size of the parts of the signal we wanted to classify. Due to a rounding process in the code used to translate mk3 markings in a signal, the minimal duration any marking could have in theory in the marking signal was one second. Since there is no standard length for any given event, this seemed like the reasonnable option if we wanted to be sure to include all events referenced in our data. However, it quickly appeared that this choice was introducing too little data to efficiently classify anything. We therefore needed to introduce some context, some additional features to the models.

These features would take the form of the parts of the signal that directly preceded and followed the initial sliding window we wanted to classify. For simplicity reasons, we will be referring the the part of the signal we try to classify as the focus window, and to the neighbouring parts used as the context windows. We first considered using 15 seconds of signal before and after the focus window, but a discussion with Dr. Senny informed us that the nature of micro-awakenings (a quick change of frequency or a sudden change of amplitude) made the signal preceding the part we want to evaluate more relevant than the following. We therefore shifted the balance toward a 25 seconds of signal preceding the focus window and 5 seconds of signal following it. We could have used more data on both sides, but we did not want to overload the model with potentially useless data. It also would have posed technical issues when considering the size that long inputs would take in memory. This approach was later confirmed to be a good idea by [30].
While the use of a 1 second focus window helped being granular and detect with accuracy a suspicious part of a signal, it also introduced some noise. After all, one second of data is only 10 sampling values in an input of 310 features, as we work with a 10Hz sampling system. After further inspection of the data, it became obvious that micro awakening, while not formally being defined as such, always lasted at least 5 seconds. This allowed us to choose a focus window size of 5 seconds instead of 1, which made the part of the input defining the class more imposing, as it now constitutes 1/7 of the input instead of the previous 1/31. Our final sliding windows are therefore constituted of a 25 seconds long left context window, then 5 seconds of focus window, and finally 5 additional seconds of context window.

Once the sizes of the various parts of the inputs were defined, we needed to decide what would define the class of an input based on the values of the marking signal of the focus window. Of course, if there is only one kind of class in the focus window, this will be the associated class. But what happens if we get a focus window overlapping on two or three different classes ? We first decided to exclude any sample that would fall into this case, to avoid ambiguous Y values. But doing so excluded quite a lot of samples from the training, testing and validation sets, and prevented us from accurately predicting micro-awakenings in some places.

To illustrate this, we take a look at the example provided at the Figure 5.4. Here it can be seen that the Y vector portion of value 7 (indicating a zone of micro awakening) goes a bit beyond the borders of the marking. This means that the 5 seconds long focus windows that defined what class

would be assigned to each 5 seconds segment overlapped on both sides of the positive zone. The focus windows therefore had both the sleep and micro awakening class represented in them. Had we applied the rule that said that a focus window is invalid if more than one class was found inside, we would have ignored this whole zone, and lost two good training samples.

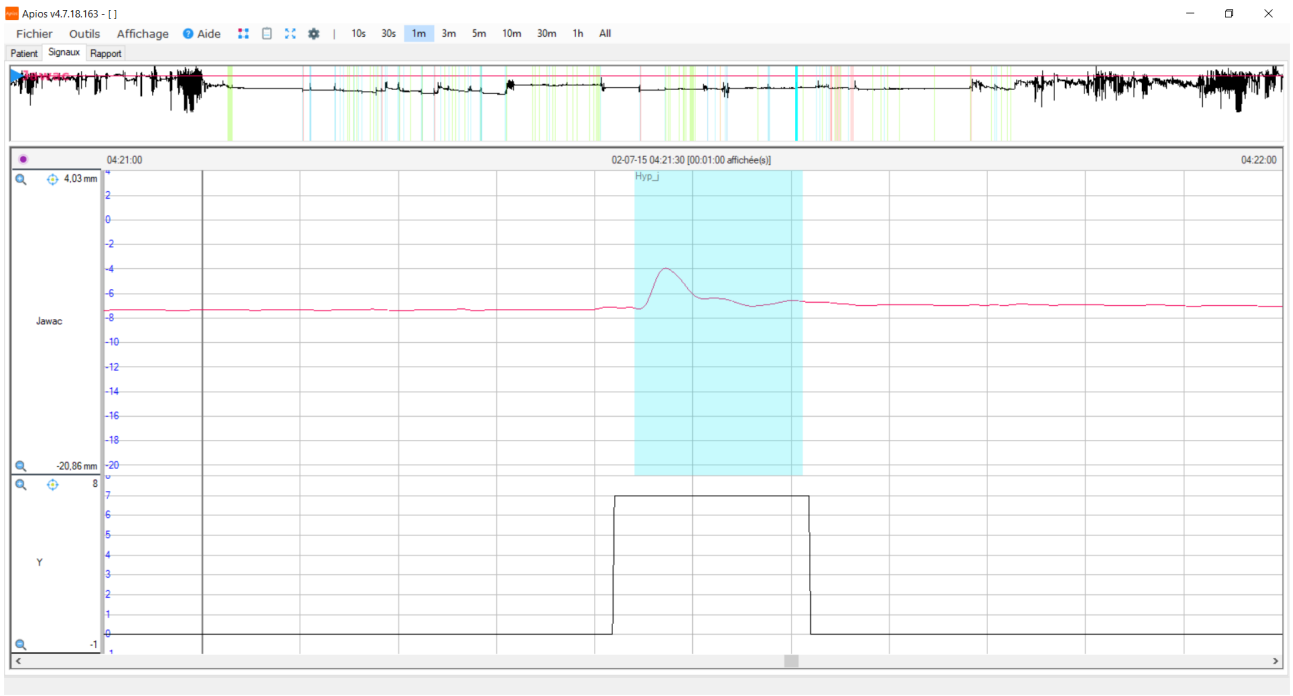

Figure 5.4: Example of valid windows that would have deleted two good positive cases examples with stricter valid windows rules

We therefore decided to simply take the majority class of the focus window, which doubled our number of positive testing samples.

We also had to decide how we would treat the classes that were neither sleep nor micro-awakening. It is frequent for all incidents to cause some sort of micro-awakenings, which means that considering these samples as simple sleep could lead to some confusion, as we could see two very similar portions of signals being classified as micro-awakening and hypopnea respectively for example. However, simply rejecting these portions of the data and not using them at all could also prevent the models from learning how to differentiate these events from micro-awakenings, leading to more prediction errors down the road. We therefore decided to treat all events as sleep, except for micro-awakenings, of course, and "Musculatory Events". These events are considered to be globally linked to micro-awakening and can therefore be seen as such.

Finally, we had to decide what to do with the "Out of Range" and "Dubious markings" labels. Given the nature of the "Out of Range" sections, we decided to simply reject any sliding window that would contain any part of these markings. This is not applicable to "Dubious Markings " though. Indeed, these signals are not unusable, we simply do not trust the annotations made on them. They can still be used as global context. We will therefore consider valid any sliding window that does not have any "Dubious Marking" in its focus window. This way, we allow these parts of the signal to be useful as additional features for an input while preventing suspicious labels to be taken as training or testing examples of our dataset.

5.3.2 Original Dataset

The way we generated our first dataset was the following : as we mentionned, we had a sliding window of 35 seconds sliding through the markings signal. This sliding window was constituted of a context window of 25 seconds, followed by 5 seconds of focus window and another 5 seconds of context window. It started at the beginning of each markings signal and then moved of a given amount of time steps toward the end of the signal. The amount of time steps the sliding window moves over at each iteration of the process is called the sliding size and is here fixed at $5 \times Fs$, where Fs is the frequency of the Jawac signal. In other words, the sliding window is moved of 5 seconds toward the end at each iteration. If we desired to gather more data, we could reduce this value, to create an overlap between two successive focus windows.

At each of these iterations, the sliding window is analysed to determine whether or not the sliding window is considered valid or not. When a sliding window SW is considered valid, the equivalent portion of the Jawac signal is sampled to serve as a X value and is therefore added to the X vector of the training set, the validation set or the testing set, depending on which set we were currently building. Simultaneously, the Y value associated to this X sample is determined by looking at the majority class found in the interval defined by the focus window's bounds in the markings signal. The Y value is then added to the corresponding Y vector just like for X. The sliding window is then moved

An illustration is shown in Figure 5.5. The signal $AIOutput$ represents the probability of being a positive the AI associates to a sample. We represent on this signal samples that were ignored by the system by negative values, purely for visual convenience. We can see that the zone marked as dubious corresponds to an ignored segment of the signal.

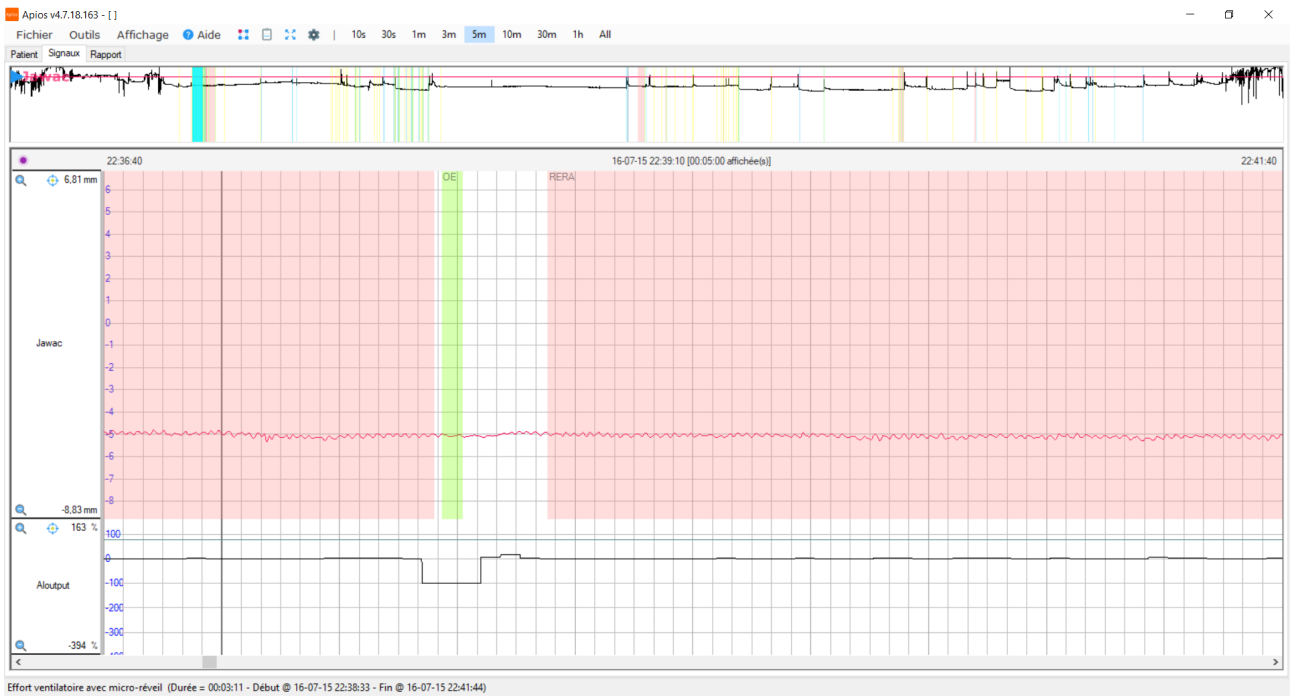

Figure 5.5: When a focus window has dubious markings in it, the sample is not taken into account by the predictions/training algorithm

We use recordings 1 to 65 for the training set, 66 to 81 for the validation set and 82 to 95 for the

testing set. This gives us the following table for the data repartition.

Data repartition in the original dataset				
Sets	Training Set	Validation Set	Testing Set	Total
Sleep	252 508	60 175	57 465	370 148
Micev	7 803	2 167	1 613	11 583
All classes	260 311	62 342	59 078	381 731

5.3.3 Simplified Dataset

After some experiences with our basic original dataset we noticed that while our predictions when visualized on Apios seemed to be relatively coherent, the results from our metrics were not telling the same story. We had indeed much more errors and lower recall and precision rates than expected. We soon realized that very often, our models would fail to predict positive samples that were pretty hard to classify as such. Indeed, some regions marked as positive cases by the doctor were not differentiable from a classic negative region.

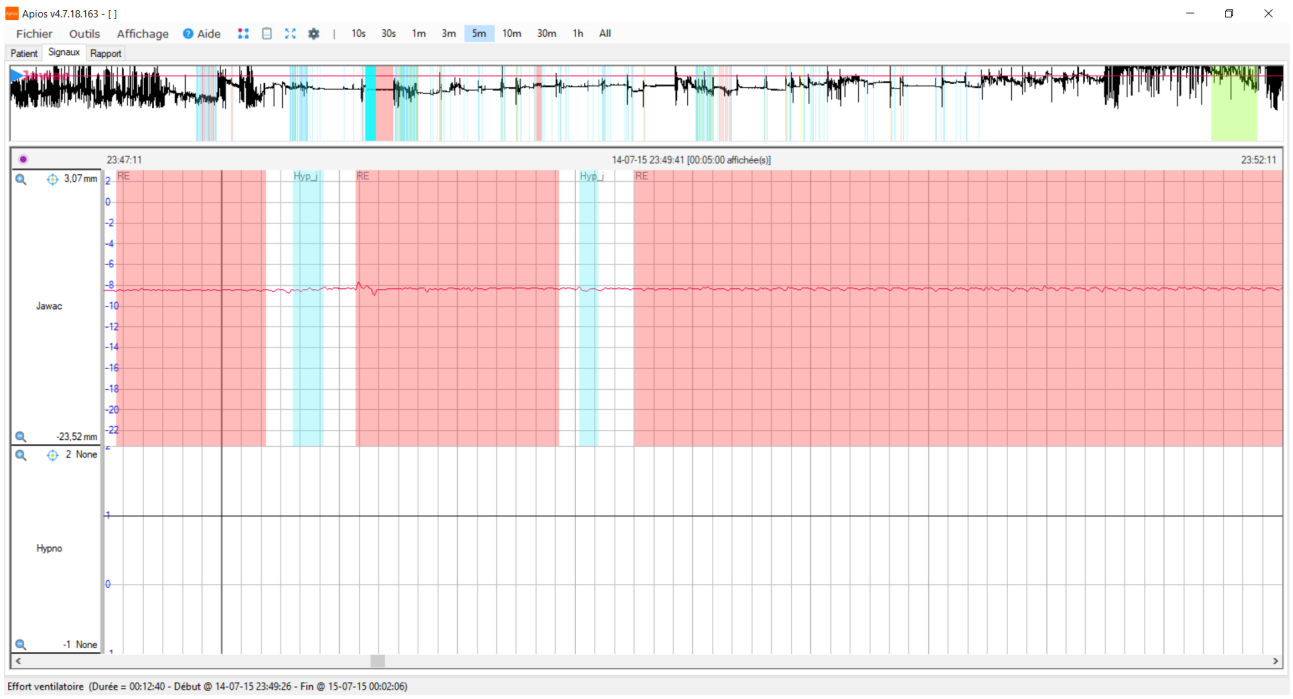

Figure 5.6: Positive markings of the doctor indistinguishable from negative zones

As we can see in the image (5.6), the Jawac signal marked as a micro-awakening presents no sign of an actual nervous arousal. It must be noted that the doctor had a variety of various signals at his disposal to make his diagnosis, which means that he could very well have some additional informations that our single Jawac signal can't translate that could explain these irregularities. They however can also can be linked to debatable human intepretation ([38, 39, 40, 41] showed that different sleep specialists can score various sleep events and sleep stages with high variability not only with other practicians but also with themselves.), or to the use of threshold in the decision making of the doctor, who is after all trying to put binary labels on physiological events that are not binary.

In an effort to get rid of as many external bias as possible, we decided to create a second dataset based on modified EDFs. We took back our original files and had a qualified Nomics employee review each individual recording to identify individual markings that sounded dubious. These markings were then changed so that the region they delimited would not be seen as "Micro Awakenings" anymore, but " Dubious Markings". This would allow us to keep these parts of the signal as context for the context windows of the focus windows, while avoiding their use in the determination of the class of a sample. In other words, our models could no longer be trained on samples based on these regions and our models would no longer be evaluated on their predictions on these regions. That way, we avoid confusing their learning process by presenting them samples qualified as positives but with all the characteristics of a negative, and we would not introduce a bias in our evaluations by penalizing the models for predicting such samples as negatives. The Figure 5.7 is the same signal segment as in Figure 5.6 but after the modifications explained above. The dubious markings were changed accordingly.

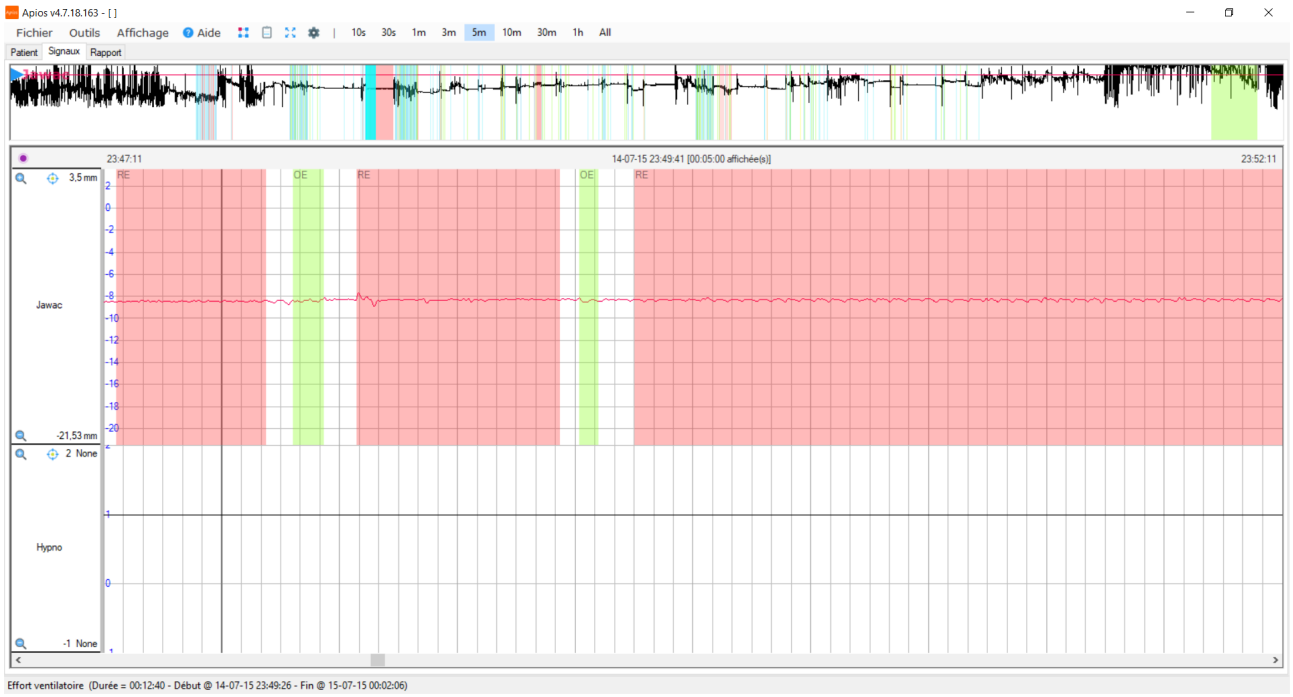

Figure 5.7: The suspicious markings of the doctor were replaced by "dubious markings"

On the basis of these new EDF files, we built a new dataset, in the exact same fashion as we did for the original dataset. The training set, validation set and testing set are based on the exact same files as before. The class distribution of all sets are presented in the table below.

Data repartition in simplified dataset				
Sets	Training Set	Validation Set	Testing Set	Total
Sleep	238 055	56 779	53 258	348 092
Micev	7 362	2 071	1 428	10 861
All classes	245 417	58 850	54 686	358 953

5.3.4 Senny filtered Dataset

After experimenting with our simplified dataset, we noticed that our models still encountered problems originating from the dataset. Indeed, it was pretty common to have regions of the dataset that were not labelled as positive cases but that presented all the characteristics of one. Such a zone is illustrated in Figure 5.8

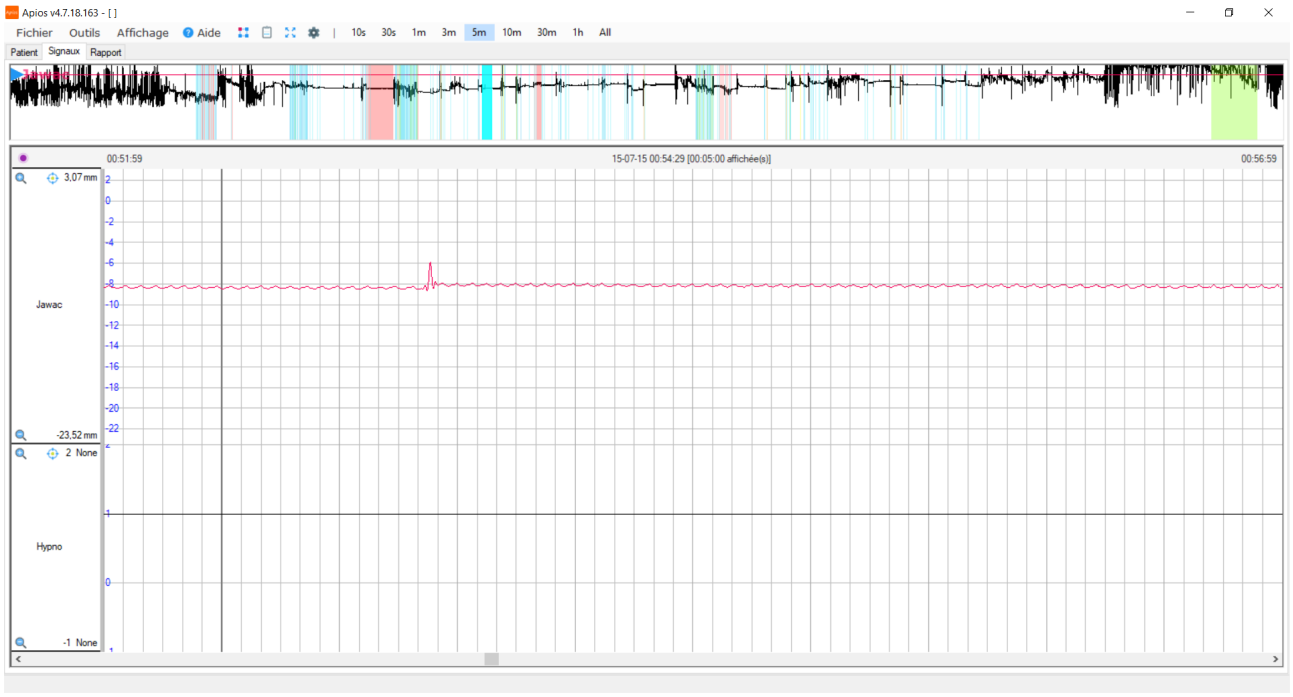

Figure 5.8: Zone marked as negative but presenting all the characteristics of an arousal

These zones were often classified as positives by our models, and while it was technically an error, we could not really blame the model for it as these kind of samples were exactly the ones we aim to classify as positives. We had these situations reviewed by Nomics employees, who acknowledged that it was indeed not a problem if the model classified these regions as positives.

It still was problematic for these predictions to be seen as errors, as it impacts our capacity to correctly evaluate our models and therefore acts as an obstacle in our research of an optimal model.

The solution we brought was the use of the algorithm of Dr. Senny as a filter : the prediction algorithm of Dr. Senny would be run on all files of the dataset, and the predictions outputed would be compared to the labels of the doctor. The idea was to mark as dubious the parts of the signals that were considered positive by the algorithm of Dr. Senny and not by the doctor. That way, we ignore the regions that are quite obviously positives that were ignored in the diagnosis of the doctor for the training as well as for the evaluation of the models, while not adding any bias. We do not label them as positives to avoid teaching our models to imitate the algorithm of Dr Senny itself. The same signal segment as in Figure 5.8 is shown in Figure 5.9 after the algorithm of Dr Senny was run on it. The prediction outputed by Senny's algorithm is visualized by the purple highlight on the Hypno signal.

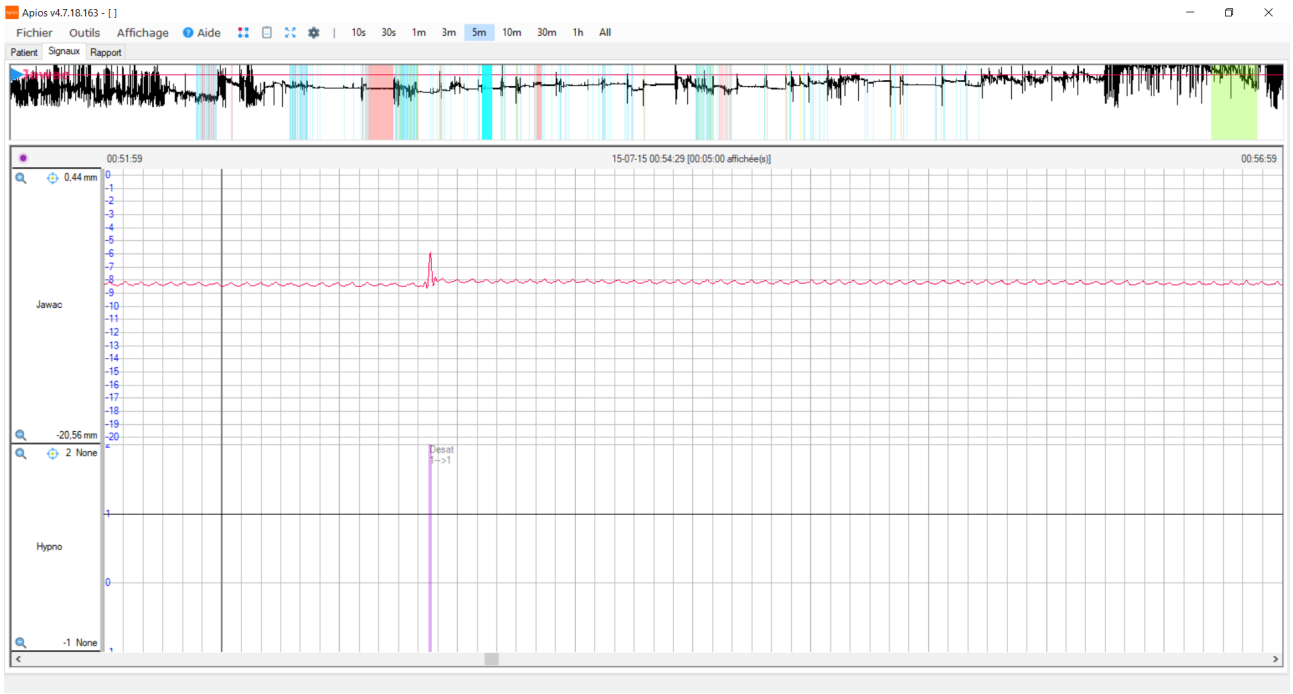

Figure 5.9: Zone marked as negative but presenting all the characteristics of an arousal is identified as positive by Senny's algorithm (purple highlight on the hypno signal)

We applied this idea by modifying the marking signal in our final EDF files by comparing it to the predictions markings outputed by Senny's work. These predictions were given to us in mk1 files, a format similar to mk3. We could parse them into our original mk3 files and treat them as any other class thanks to our existing data pipeline. We simply ignored all markings that were at least partially overlapping with any positive, dubious or out of range marking. The remaining predictions markings were simply transformed in dubious markings after having transformed the markings into the marking signal. We then proceeded to the construction of our dataset as usual.

The class distribution of the "Senny filtered" dataset, as we will call it, is shown in the following table.

Data repartition in the Senny filtered Dataset				
Sets	Training Set	Validation Set	Testing Set	Total
Sleep	258 297	61 107	58 514	377 918
Micev	8542	2 339	1 764	12 645
All classes	266 839	63 446	60 278	390 563

5.3.5 Dataset simplified and Senny filtered

We had better results with both the Senny filtered and simplified dataset. However, both strategies tackle different issues : the Senny filter allows us to avoid predicting False Positives that should be seen as True Positives and the simplified dataset helps us avoid predicting False Negatives that should be seen as True Negatives.
We therefore combined both approaches and built a "Simplified and Senny filtered dataset". This dataset is as close as we can get to an unbiased estimation base for evaluating our models, as well as an unbiased training set. Its class distribution is displayed below.

Data repartition in the simplified Senny filtered dataset				
Sets	Training Set	Validation Set	Testing Set	Total
Sleep	251 981	59 802	57 968	369 751
Micev	8 176	2 387	1 619	12 182
All classes	260 157	62 189	59 587	381 933

While this dataset certainly helps us have a more honest estimation of our models performances, it is still plagued with a couple issues : the corrections made by Nomics employee are also man made, and therefore vulnerable to human errors, as it is a tedious and repetitive task. The employee also did not benefit from the multitude of additional infos the doctor had. On the filter part, while the algorithm certainly provides a consistent and somewhat reliable way of eliminating suspicious samples, it is still not perfect in itself and will make mistakes. While it is not really problematic to lose a few negative cases examples that would be misqualified as positives, not predicting as positives suspicious zones that were not labelled by the doctor is. Unfortunately, there is no way for us to really enhance this pre-processing. We therefore have to accept that there will be a certain degree only to which our evaluation metrics will be accurate.

Chapter 6

Model architectures

The machine learning part of this project was done on Python 3.7.6. We mainly used sklearn for our out of the box models and used their metrics toolbox for all scoring steps, unless specified otherwise. The Deep Learning framework used was Pytorch.

6.1 Random Forest

We first trained a Random Forest with 500 estimators and the default options present in sklearn. This first model has the role to provide us with a baseline to beat with our next models, and confirm the usability of the data.

In the following table, we can see the results obtained when training our basic Random Forest on all datasets. The results obtained are always based on the predictions of the different models on the testing set of the simplified and Senny filtered (SSF) dataset, to provide a common comparison basis. We also had to lower the threshold above which we consider the output of the Random Forest to be a positive prediction. We defined the best threshold as the one that would maximize the F1-Score. We found the threshold 0.17 to correspond to this definition.

Results of predictions on the SSF testing set for a RF with 500 trees			
Training Set of the RF	Precision	Recall	AUC
Original	0.62	0.68	0.667
Simplified	0.61	0.71	0.678
SSF	0.63	0.7	0.694

The Precision-Recall curve of the best RF model according to the AUC in the normal system is shown in Figure 6.1

These results however suffer from the problems highlighted in Chapter 3. For example, a correct prediction of a positive case surrounded by 2 samples also labeled as positives but not predicted as such will be counted as 2 errors for one success in the classic metrics. We therefore also computed the results expressed in our home made Jaccard metrics. We arbitrarily decided to set the Jaccard threshold to 0.3 (see section 3.3.3).

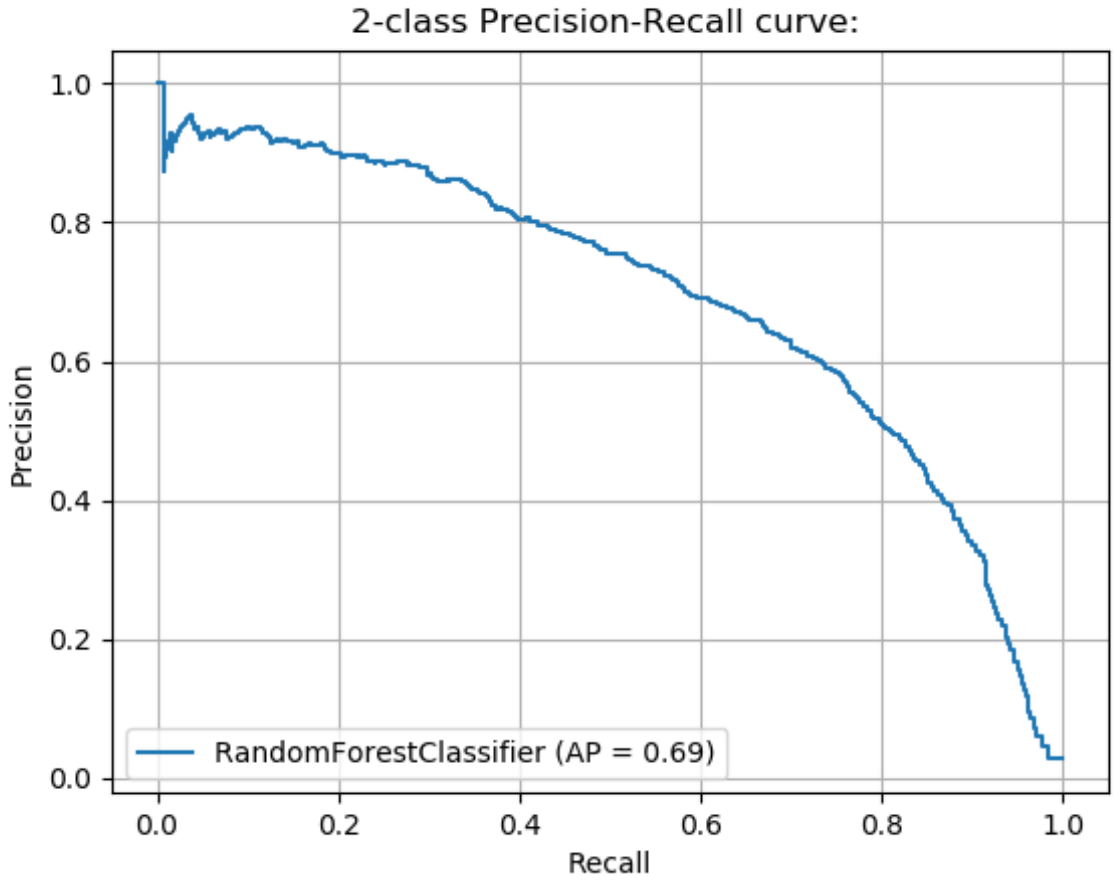

Figure 6.1: RF of 500 trees trained and tested on a simplified and Senny filtered dataset

Jaccard Results of predictions on the SSF testing set for a RF with 500 trees			
Training Set of the RF	Precision	Recall	Best F1 Score
Original	0.7229	0.6756	0.6985
Simplified	0.7273	0.6825	0.7042
SSF	0.7303	0.6912	0.7102

We used these results as a benchmark for evaluating the performances of our next experiments. We tried to enhance these results by performing a grid search on 2 parameters : the amount of trees and the amount of features the best split is decided upon. We tried to make the number of trees vary for values between 500 and 1000, with increments of 100. The AUC on the normal PR curves were always either very similar or simply worse than for 500 estimators. We therefore decided to keep this amount of estimators. Concerning the max features value, we tried the two values recommended by sklearn : the log2 of the number of features and the square root of the number of features. Log2 performed significantly worse than the square root system, we therefore used the later. Our final Random Forest therefore has 500 trees and is trained with a max amount of features equal to the square root of the total amount of features available. The performances of this RF are the ones displayed here.

Once we got good enough results with the random forests, we decided to move onto the fully connected neural networks, as we now had exploitable data and a benchmark to judge new perfor-

mances on.

6.2 Fully Connected Neural Networks

As Random Forest gave results that were rather encouraging, we tried to build more and more complex fully connected architectures, to try to improve results. While the results were often on par with the ones we obtained with the random forest, we still obtained a couple networks that performed significantly better.

The networks were built with an iterative process : we started with a very simple network with no hidden layers or any additional features. We then worked our way up, adding layers and trying to always improve from previous experiments, adding sophistication as the results came.

At first, the networks were not learning well. We realised that depending on some hyperparameters, the network would present either a strong bias toward predicting micro awakenings, or toward predicting sleep. The loss was also rather inconsistent from one batch to another. We guessed that it probably came from the unbalanced character of our dataset : our batch size being of only 128, it was likely that from time to time, we would get no arousal sample in a batch, or on the opposite, 4 or 5 of them, leading to a bit more error than usual. To tackle these issues, we used a RandomWeightSampler. It is a tool available in Pytorch allowing the attribution of probabilities to each sample of the dataset. Each batch is then filled by randomly selecting samples with the probability attributed to them. We here attributed to each positive sample a probability that made them twice as likely to be selected as negative samples. This helped the network learn to differentiate positive classes from negative classes without suffering from the unbalanced class distribution. We used this technique for both fully connected models and convolutional ones.

During the training of neural networks, overfitting can happen. This happens when the training loss keeps going down but the validation loss stops diminishing and starts going up. This means that the NN is not learning anymore but is actually memorizing the training set. To counter this, we evaluated the model on the validation set at each epoch, and kept in memory what was the current best validation loss obtained. If the model at the end of an epoch got a better validation loss than the current best, we saved that model. At the end of the training, we end up with the model that performed best on the validation set, not on the training one. The same procedure was applied for the training of CNNs. Note that the use of dropout was also attempted to help against overfitting but led to worst results.

We here present the results of the best of the fully connected models we ended up with, that we called FC4. Once again, the model has been trained on several versions of our dataset and evaluated on the testing set of the SSF dataset. We selected the best Fully Connected model by comparing the AUC of different models to each other. When a promising model comes up, we do a more granular approach and look for its maximum Jaccard f1_score. The best model is then the one that manages to get the best Jaccard f1_score on the SSF testing set.

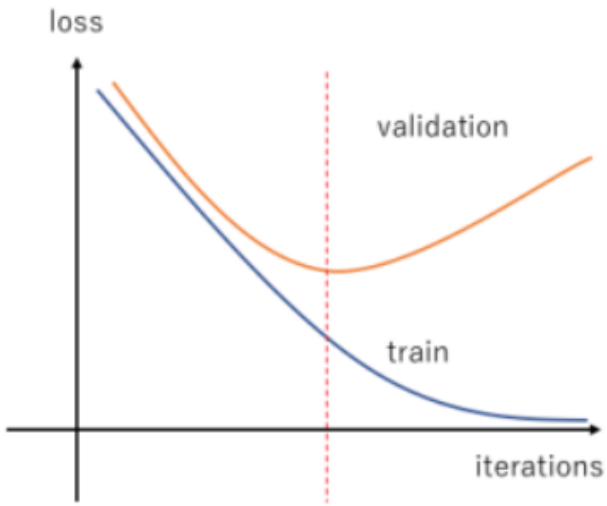

Figure 6.2: A deep learning model learning too long can have its training loss keep decreasing while the validation loss stop decreasing : this is overfitting and means the NN is not learning to generalize anymore. The best model is the one with the lowest validation loss.

Results of predictions on the SSF testing set for FC4			
Training Set of the FC4	Precision	Recall	AUC
Original	0.66	0.64	0.599
Simplified	0.66	0.70	0.7209
SSF	0.66	0.72	0.71604

As before, we also evaluated our model with our home made metrics. For the consistency of our comparisons, the Jaccard threshold is still set at 0.3.

Jaccard results of predictions on the SSF testing set for FC4			
Training Set of the FC4	Precision	Recall	Best F1 Score
Original	0.7251	0.67	0.6965
Simplified	0.7651	0.6942	0.7279
SSF	0.7652	0.7113	0.7373

The PR curve of the best FC4 model according to the AUC in the normal system is presented in Figure 6.3

As one can see, this model is fairly better than our best Random Forest, managing a maximum Jaccard f1_score of 0.737 on the SSF testing set, where the Random Forest performed a 0.7102 Jaccard f1_score at best.

The architecture of this model was as shown in 6.4

6.3 Convolutional Neural Networks

Finally, we tried to improve upon the results of FC4 by using CNNs. We started by trying to add convolutional layers to the fully connected layers constituting the FC4 model, and experimented in a similar way to what we did with fully connected models. The best CNN model was determined by

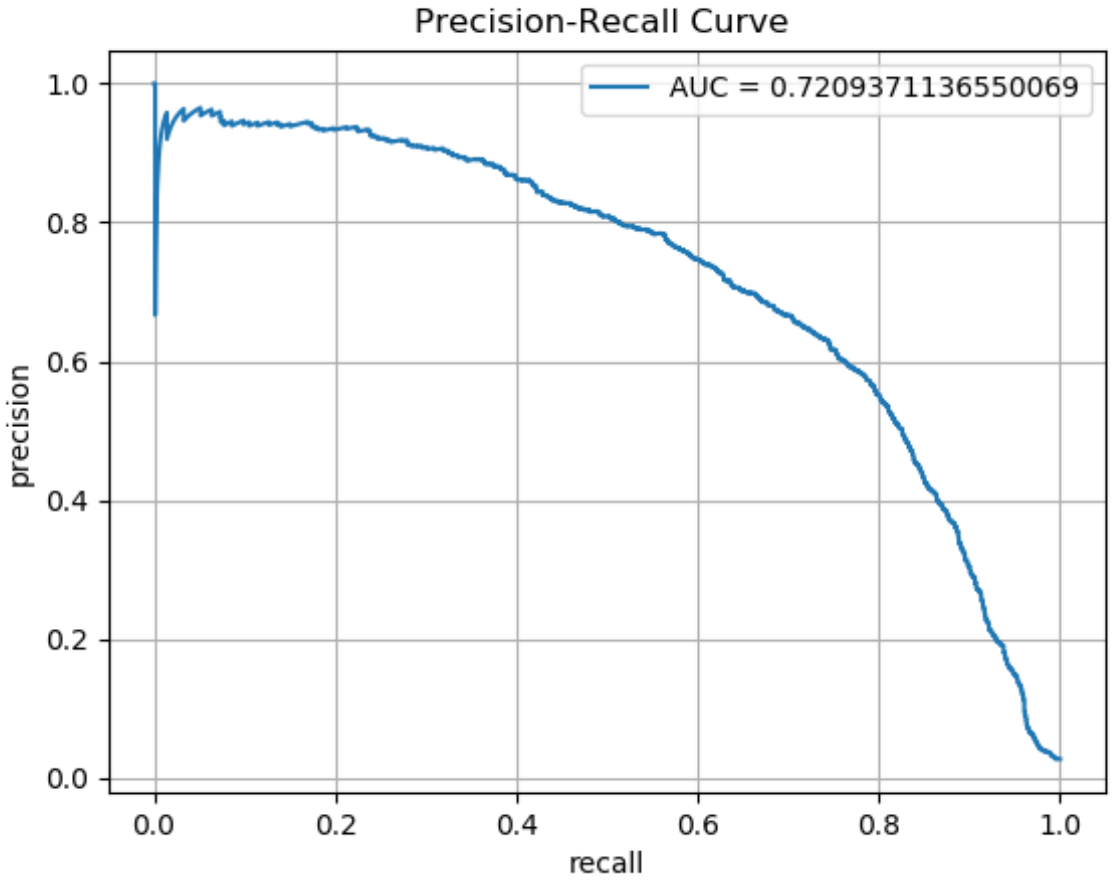

Figure 6.3: FC4 trained on a simplified dataset and tested on a simplified and Senny filtered dataset

the same process as for the FC models. We ended up with CNN2, which had results very similar to the ones we had with FC4.

Results of predictions on the SSF testing set for CNN2			
Training Set of the CNN2	Precision	Recall	AUC
Original	0.73	0.68	0.71744
Simplified	0.69	0.69	0.69165
SSF	0.67	0.67	0.67177

Here are the same results in Jaccard fashion, with a 0.3 Jaccard threshold :

Jaccard results of predictions on the SSF testing set for CNN2			
Training Set of the CNN2	Precision	Recall	Best F1 Score
Original	0.8091	0.6842	0.7414
Simplified	0.7795	0.7062	0.7410
SSF	0.7945	0.6624	0.7225

The corresponding precision recall curve shown in Figure 6.5

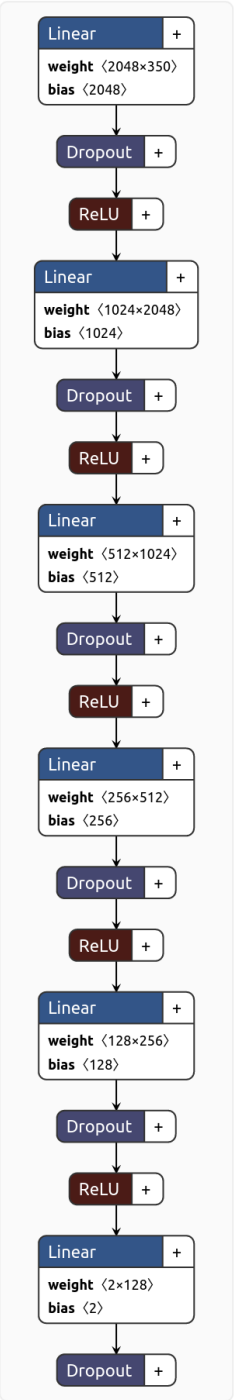

Figure 6.4: The architecture of the FC4 model

Surprisingly enough, this model is the only one on which the best performance is not obtained by training it on the SSF dataset. It reaches a maximum f1_score of 0.7414 on the SSF testing set, improving over the best FC model.

The architecture of CNN2 is as showed on figure 6.6 :

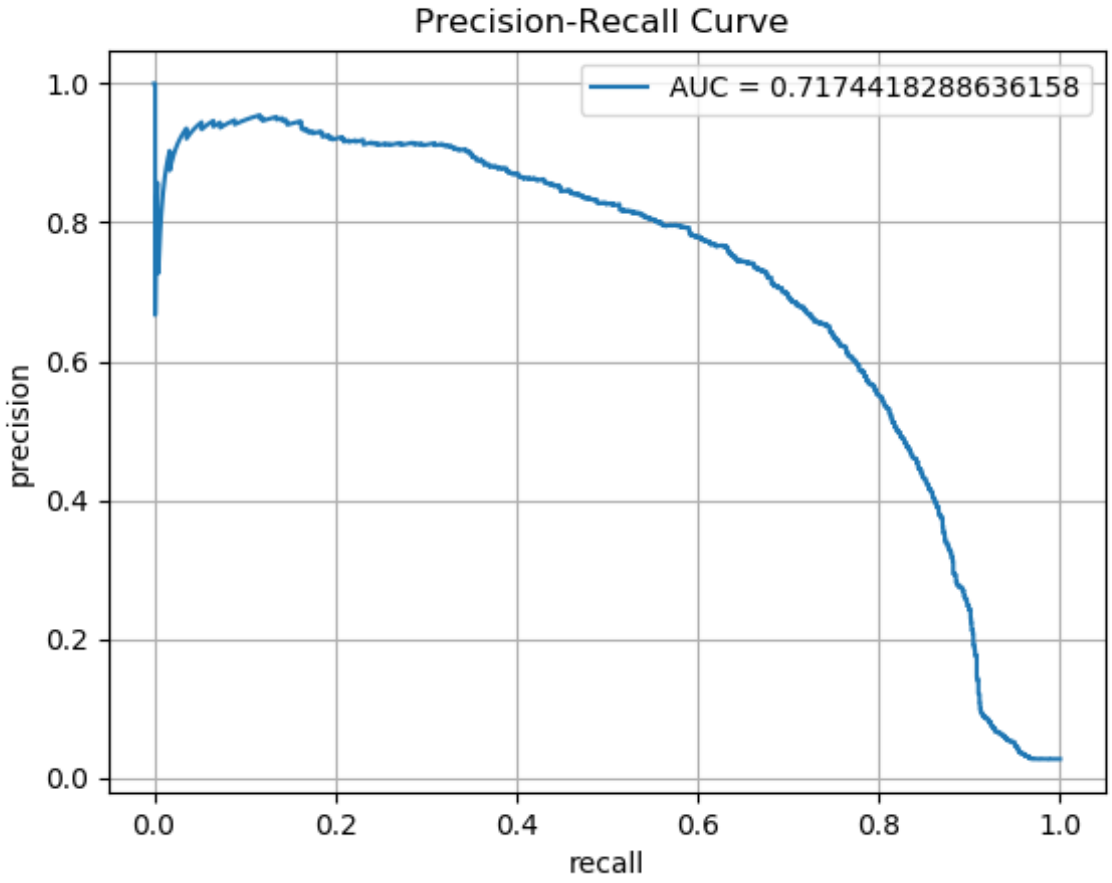

Figure 6.5: CNN2 trained on the original dataset and tested on the SSF dataset

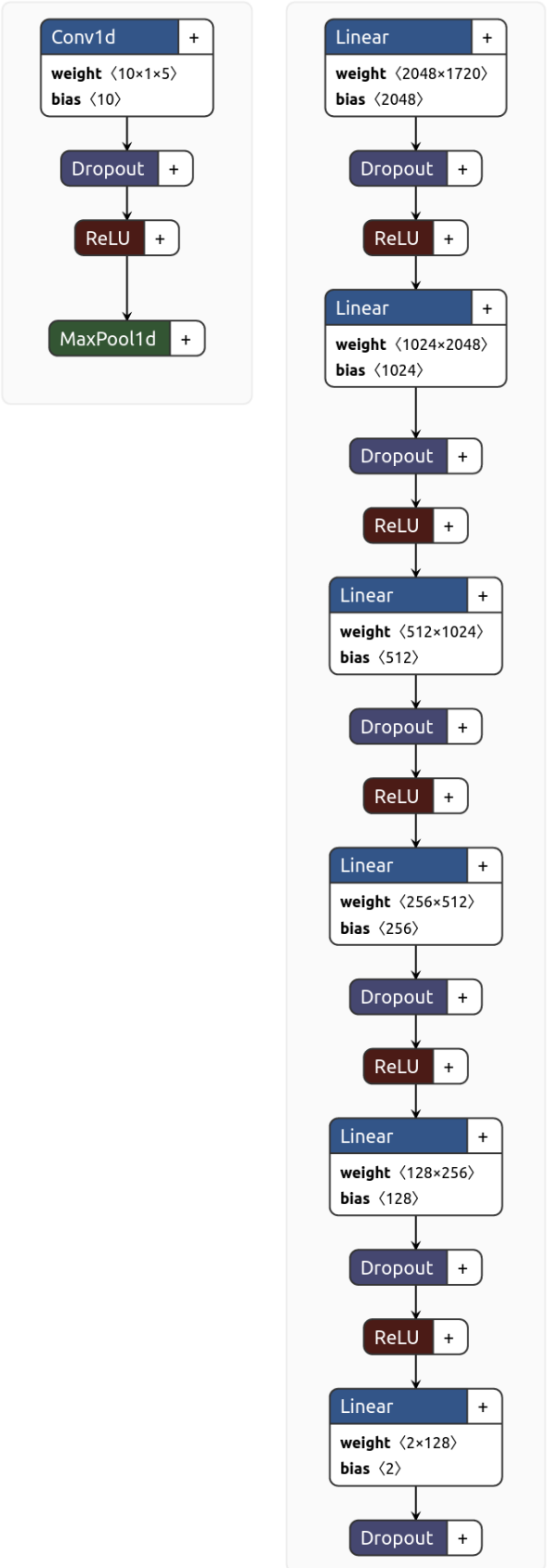

Figure 6.6: CNN2 model architecture

Chapter 7

Results discussions

7.1 Best model

From the results gathered in the last section, we fairly quickly realize that Random Forest perform objectively worse than Neural Networks. Indeed their best performances, when trained on a SSF dataset, reach a Jaccard F1-score of 0.7102, whereas FC4 reaches 0.7373 when trained on the same training set. CNN2 tops all of them with a maximised Jaccard F1-Score of 0.7414. Interestingly enough, this best result is obtained with CNN2 trained on the original dataset, unaltered. This difference between NN models and RF models is also translated in a PR curve with an AUC that does not get over 70% for the RF, while CNN2 and FC4 manage to reach 71.74% and 72.09% respectively. The superposed curves of these best models are shown in Figure 7.1. It is interesting to observe that despite having a lower AUC on the PR curve than FC4, using the Jaccard system of evaluation reveals the CNN2 model to be more efficient. The PR curves of these models in the Jaccard system is shown on Figure 7.2, where we can see the edge that neural networks have over Random Forests and the very close performances between FC4 and CNN2.

7.2 Comparison of the best model with Senny's algorithm

While we decided to evaluate all of our models on the same testing set, the testing set simplified and filtered with Senny's algorithm (called the SSF testing set), it did not seem appropriate to evaluate Senny's algorithm on the same ground. Indeed, this testing set literally removes the erroneous positive predictions made by Senny's algorithm from the testing set, giving it an unfair advantage.

We therefore needed a fairer testing ground for comparing our performances. We decided to use the simplified testing set. We also lowered the Jaccard threshold to a value of 0.1, because Senny's algorithm predicts very small windows, often as small as 1 seconds, and it would be unfair to his algorithm to use bigger thresholds to qualify successes and failures. On this basis, we obtained the following results :

Jaccard results of predictions on the Simplified testing set for CNN2 and Senny's algorithm			
Model	Precision	Recall	Best F1_score
Senny	0.6065	0.7153	0.6564
CNN2	0.8435	0.6832	0.7549

Compared to Senny's model, CNN2 trades a 3% loss in recall for a 24% increase in precision, in the Jaccard system.

7.3 Discussion on the results obtained

The first thing that can be observed about the results obtained is that they are confirming the previous experiments conducted by doctor Senny : the Jawac signal gives the ability to identify micro-awakenings, at least to a certain degree. While the performances are not stellar, they still clearly indicate some learning and for the most part correspond to what a human specialist would identify as suspicious zones.

Another conclusion that comes to mind from the amount of pre-processing involved with the base data is that some limitations in the final performances had to be expected. We encountered a number of cases in which the data labeling hinted to either faulty labels or a lack of information conveyed by the Jawac signal. While the later was a given from the start, as the goal is to reduce the amount of data to predict the same events, the former wasn't immediately obvious. However, it came to our attention that this is not an ungrounded hypothesis. Indeed, a variety of studies have shown that several sleep doctors scoring the same recordings would result in frequent lack of agreement [40, 38, 41, 39]. Moreover, other studies showed that the same doctor scoring the same recordings on different occasions was likely to disagree with himself [41]. [39] even showed that on the specific case of arousals, the interclass correlation index of scored recordings made by different experts reached no more than 0.68, with an interval of confidence of [0.5-0.85]. With this in mind, it is not far-fetched to attribute some of the irregularities encountered to some debatable labeling. We show in Figures 7.3 and 7.4 some examples of debatable errors.

It is noteworthy to point out that when the AI gets the wrong output, the probability it associates to the sample (represented by the signal AIoutput on Apios) evaluated does seem to correlate with the magnitude of the event. Indeed, one can see on the Figure 7.5 that the AIOutput signal does get excited when the Jawac signal presents signs of arousal, and is reliably flat when the Jawac signal is as well.

Despite this, the results do bring some satisfaction. We have shown that on a same testing set with same Jaccard coefficient threshold, we can get a significantly better precision than the one obtained by Senny's algorithm for a small loss in recall. When tested on the best testing set we could provide, we reached a precision 24% better for a loss in recall of 3%.

We tried to understand where the errors the model was making came from, by trying to identify some pattern from the worst diagnosed recordings, in the hope that we could again tweak the data to better the results. We therefore plotted the PR Jaccard curve for each individual recording, and obtained the graph showed in Figure 7.6. We observed that 4 recordings in particular seemed to behave worst than average. We confirmed the difficult nature of two of these by having a look at the same plot for the RF and FC4 models. Patient 85 and 81 always gets sub-par performances while patient 89 is always one of the 4 worse diagnosed recordings. We therefore took a look at the predictions of our model on these three patients.

We noticed right away that while the vast majority of the patients have between 60 and 140 predicted micro awakenings per night, patient 85 had only 16 while patient 89 had over 200. Upon further inspection, it appeared that patient 85 had actually a very calm and uneventful night, only interrupted by a handful of awakenings and a couple SDBs here and there, but hardly enough to be concerning, according to Nomics. The poor performances came from a mix of a few genuine errors from the model, 5 micro awakenings predicted right before the patient actually woke up for good, and a couple positives predicted over suspicious zones that neither the doctor nor Senny's algorithm

had identified. In short, these bad results mostly came from the very low number of elements to be detected at all present in the recording combined with mostly debatable mistakes. Patient 81 had the exact same behavior. For patient 89, the scenario was the opposite : the large number of predictions were linked to a chaotic night characterised by constant effort by the patient and an abnormal amount of SDBs all night long. Since most SDB are prone to show midsagittal jaw motion with the same kind of change in frequency as arousals, it is not surprising that sometimes the model will predict zones labeled as SDBs as arousals. This explains the unusually high amount of positives predicted.

It is important to highlight two things here :

- Patient 89 had significantly better performances than patient 85 despite being the second hardest recording to diagnose, strengthening the hypothesis that the poor results of the later mostly come from the very small number of events happening in that recording.
- While the performance on both recordings were quite poor, if we take a step back and put this back into a practical context, the main goal here would be reached. In a real setting, patient 85 would be considered healthy and our model would have predicted such a low amount of micro awakenings that the right conclusion would have been reached. The opposite is true for patient 89, which had a night filled with SDBs and was predicted an unusually high amount of micro awakenings, which would have definitely triggered furthers diagnostics.

Finally, one must consider that the significant difference between the results showcased in this work compared to the one obtained in state of the art literature has to be put into context. In the reviewed papers, the presented models had to learn to imitate human diagnosis on the same signals that were used to set the labels. It is not the case here, where the definitions and human decisions were made on the EEG and the networks had to learn to predict these diagnosis on the basis of the Jawac.

Of the unusual look of the individual Jaccard PR curves

The unusual look of these curves come from several factors related to the Jaccard threshold system. We observed that most of the time the probabilities outputs on a given recording tend to only take a handful of different values. This means that for small variations in the threshold used for predictions, the amount of newly positive predicted samples can stay at zero for some time. Similarly we can get for a small variation in the threshold a sudden pack of new predicted positives as we reach one of these several levels of probabilities, meeting the criteria to include a number of samples that were until then not considered positives. These sudden packs of new positive predictions lead to these observed sudden drop in the curve when we mostly get FPs and to plateaus when we mostly get TPs.
Another factor is the sheer amount of predictions made in some recordings. We observe that the curves that appear flattest and with the most unusual behavior correspond to the patients with the most uneventful nights. This makes sense : the less events there are to detect to begin with, the bigger will be the impact of a few new elements being predicted positive, would it be correctly predicted or not. As an illustration, we provide in Figure 7.7 the plot of the Precision Curve for patient 85, which fits this profile of uneventful night (16 arousals detected), and highlight on it the various Precision-Recall pairs that were computed.

One can observe that despite having computed the Precision Recall pairs for all thresholds values between 1 and 0 with steps of 0.01, only a handful of different points were identified, thus illustrating our previous points.

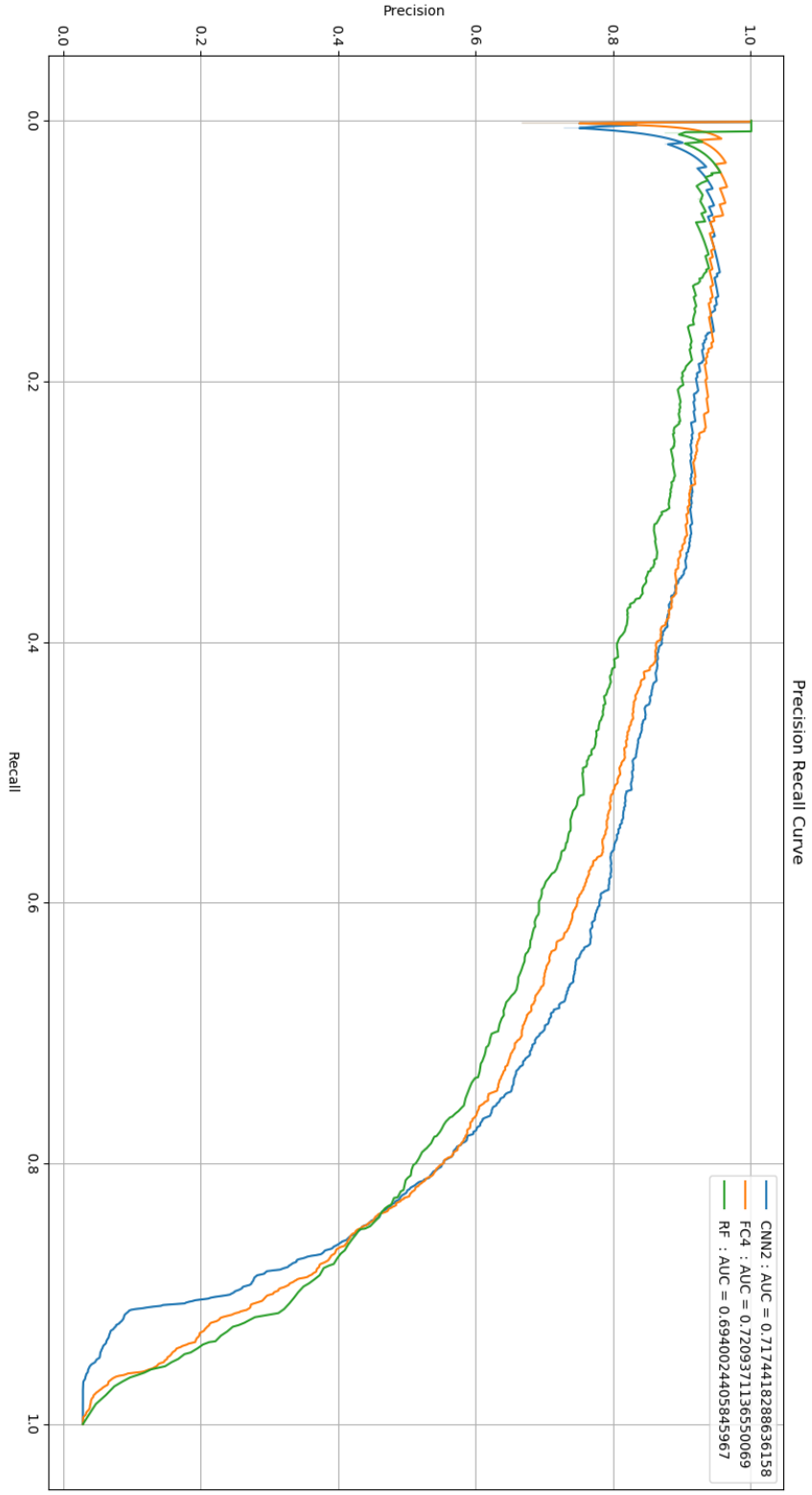

Figure 7.1: Comparison of CNN2, FC4 and RF500 Precision Recall Curves in the normal system (non Jaccard)

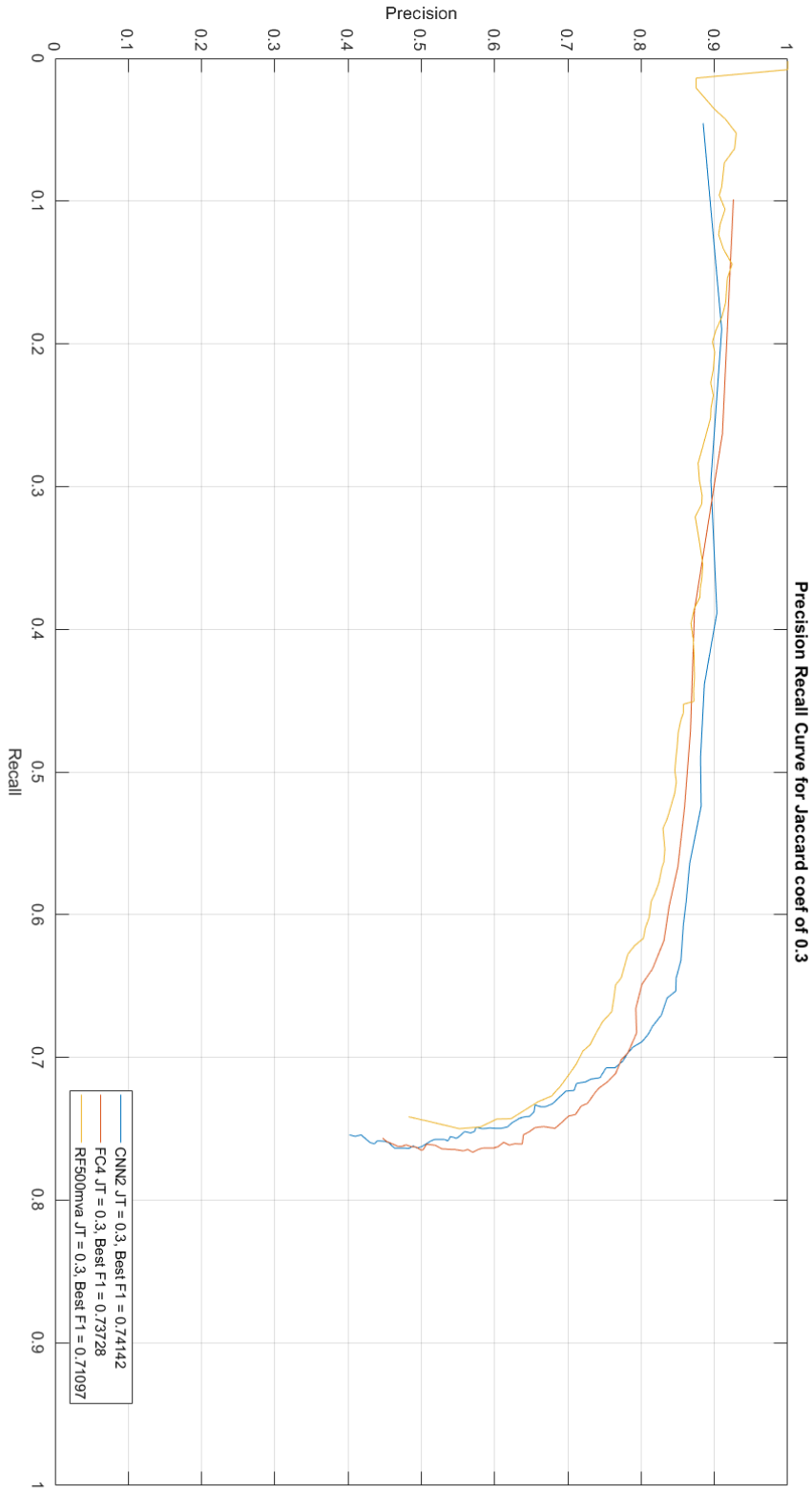

Figure 7.2: Comparison of CNN2, FC4 and RF500 Precision Recall Curves in theJaccard system

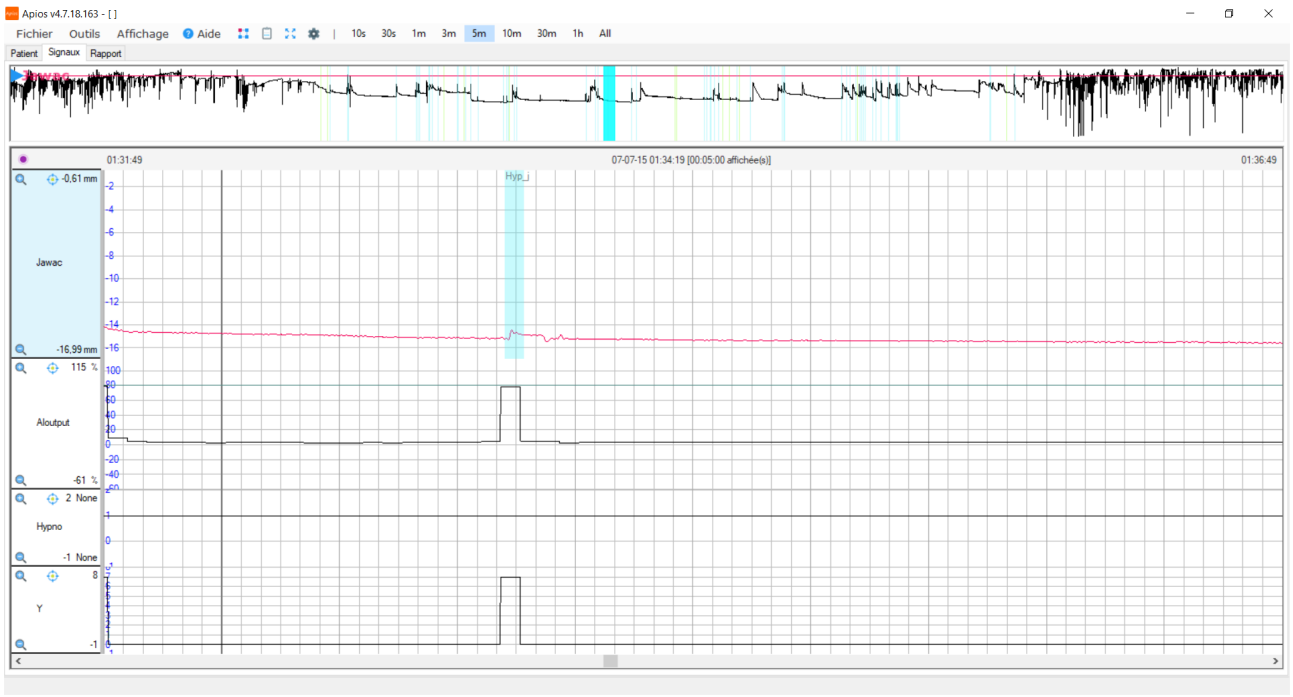

Figure 7.3: The prediction is a debatable False Negative. While there has a been a small jaw movement, it doesn't reach the 2mm amplitude of movement judged necessary to call for an arousal, yet is labeled as such. Note that the AI still noticed it, as the AIOutput signals shows a sudden excitation right where it happens, missing the needed threshold by a few percents

Figure 7.4: This is a debatable False Positive. The amplitude is of 2 mm and there is a clear frequency shift. When put in comparison with 7.3, it is hard to understand why 7.1 is considered a positive and this one not

Figure 7.5: Small movements of jaw classified as arousals are not classified as such by the AI, yet the AI noticed something, as the AIoutput signal attests.

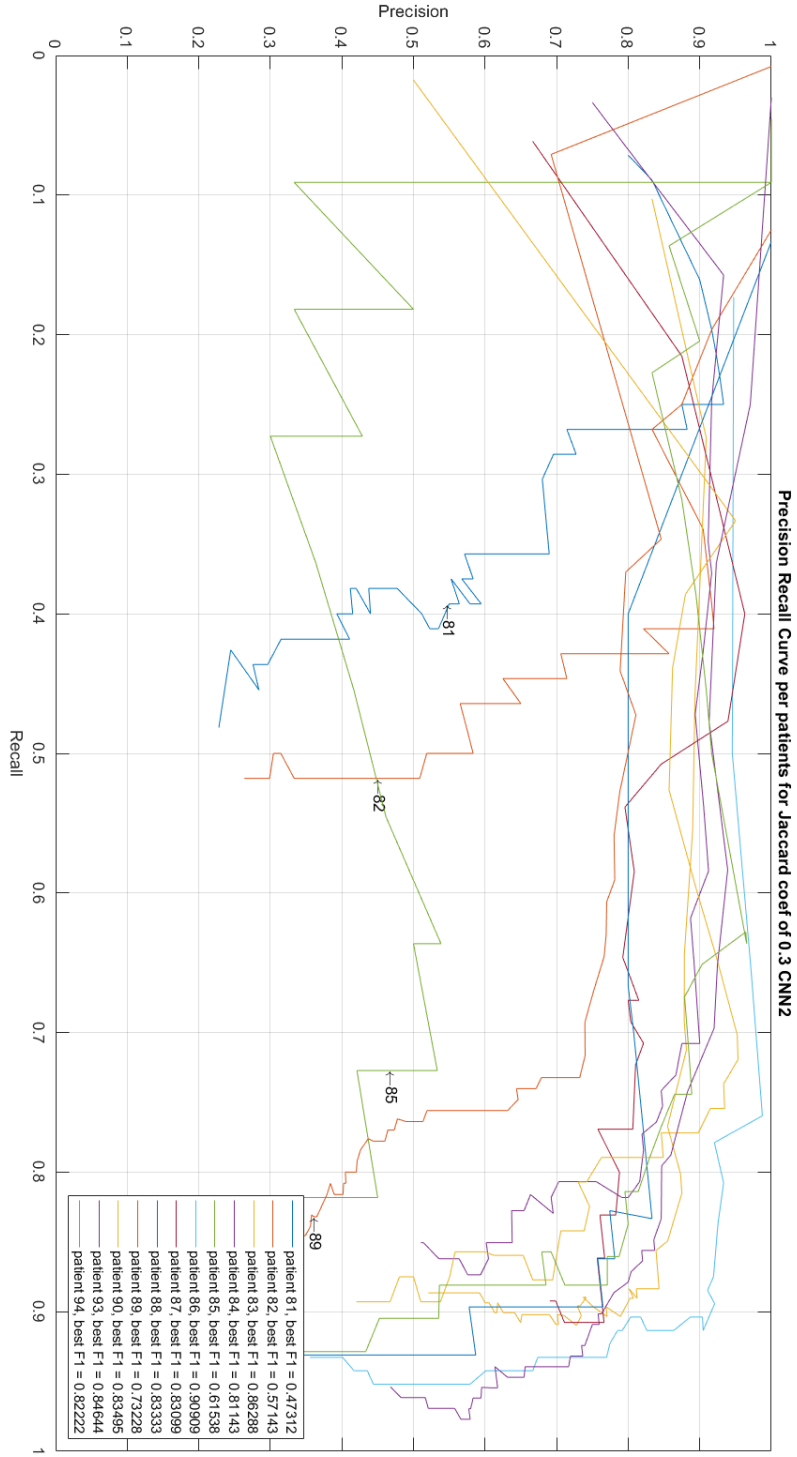

Figure 7.6: The Jaccard curve of each individual patient recording contained in the testing set based on the predictions of CNN2 trained on the original dataset. The performances are notably bad on patients 81,82,85,89 (highlighted by arrows)

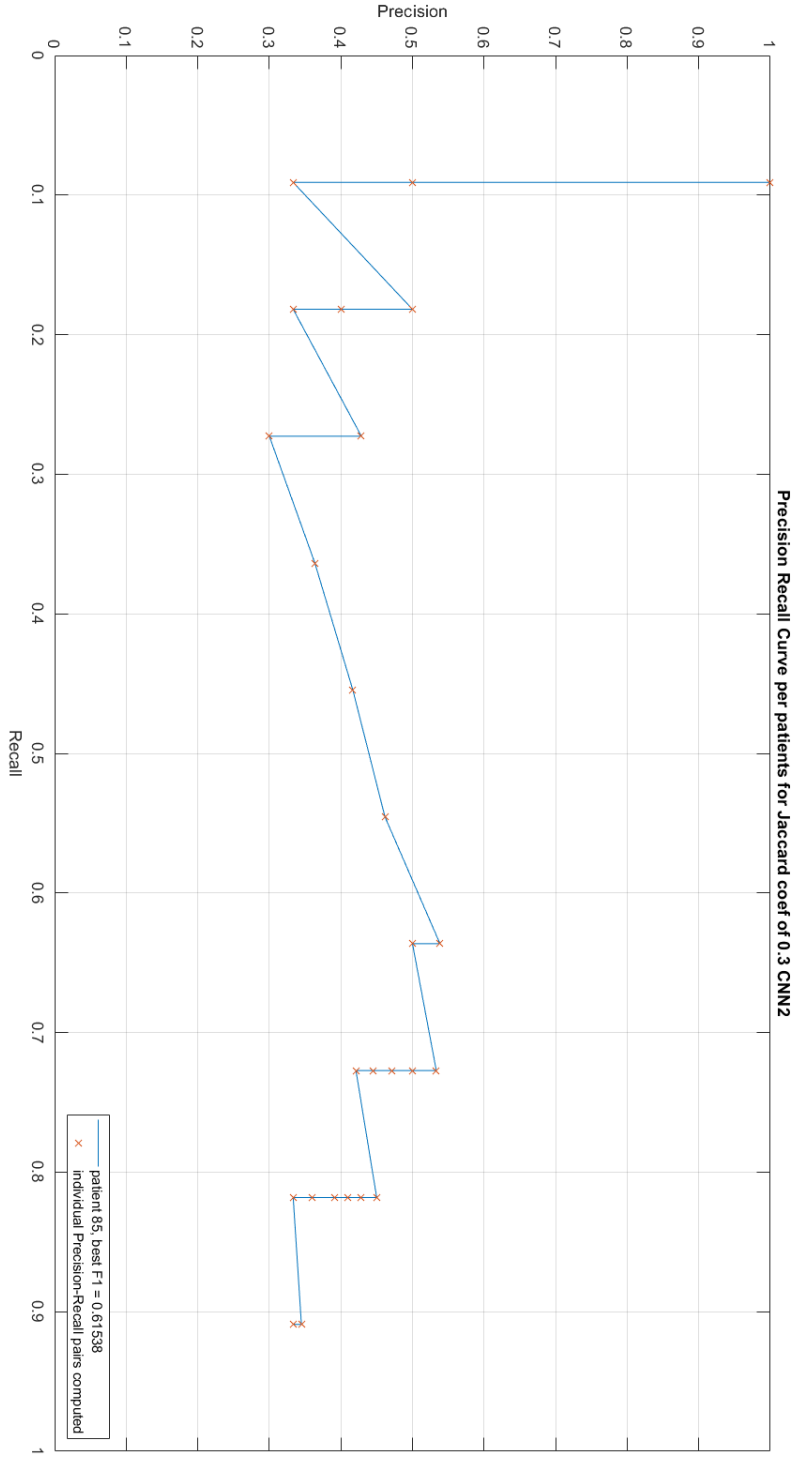

Figure 7.7: Illustration of individual points of the PR curve for patient 85. Because of the low amount of events during this specific night, a sudden pack of new predictions can have a big impact on the Precision and Recall values, leading to big gaps between two adjacent points in the curve. These explain the plateaus and sudden drops we can observe

Chapter 8

Conclusion and guidelines for future work

The work we present confirms that the Jawac signal is suited for the automatic detection of arousals and the identification of patients suffering from SDBs. While improvement was made over the last best working algorithm, there seemed to be some inherent limitations stemming from the data. From our understanding, these limitations come from the labeling of the data, which if corrected could lead to some significative improvement over performances. These improvements would come from two major factors : the learning process of the models would not be hurt by confusing samples labeled differently yet presenting the same core characteristics, and secondly, predictions deemed erroneous while admittedly justified would happen less often.

We propose the following solutions :

- Using several doctors to label the data could bring more confidence in the labels and help determine wether the suspicious labels we encountered were coming from errors or if the Jawac signal simply fails to bring enough informations to all kinds of arousals.
- Doctors use specific definitions combining multiple factors before categorizing a zone as one class or another. From this, ambiguous segment can emerge. We are trying to classify shades of grey as black or white, which leads to some apparently similar samples being classified differently, just because one biometric's value was slightly different from the other and met some threshold. We could refine the classification with a scale with more than just values. Scoring the confidence in the fact that a sample is an arousal with percentage could help be a better base with the disadvantage of being more tedious
- Nomics could label the data itself, making the Y values closer to what they expect as results to better suit their needs.
- using our algorithm, one could rule out outliers patients, like patients 82 and 89, from testing and training sets, in order to enhance the quality of the training and lower the bias in the evaluations.
- following the methodology in [35], changing the way we label the data could be interesting. We could view the data as segments of 1 minutes, and label them as positives if the segment encompass at least one arousal.

In order to have a fairer metric for measuring our performances, we implemented a Jaccard index based variation of the common Precision and Recall metrics. This proved to convey a more representative evaluation of performance. We suspect that a similar problem could occur during the training

of the neural networks models, as the loss used doesn't take into account the temporal proximity between samples and therefore is oblivious to the difference in quality between a positive prediction made in the middle of a flat signal compared to a positive prediction made one sample away from a positive labeled sample. Implementing a home made loss function that would take these considerations into account could prove to be a major help for boosting the efficiency of the training of neural networks for this task. The focal loss [42] may be worth a try in this regard. We encountered the following paper [43] that could provide inspiration for the general subject of metrics and losses adapted to this problem.

While we tried to be as extensive as we could with the CNN and fully connected networks, we did not exploit all the possibilities neural networks offer. Specifically, RNN and LSTM are architectures that are well suited for time series classification and regression tasks, and have proven effective in similar works. They could therefore bring some improvements. Other worthy experiments include stacking [44], as we have noticed that our different models do not necessarily score the same on several recordings. We are however inclined to think that architectural decisions will have lesser impacts than the previously mentioned suggestions.

Bibliography

[1] F. Senny. Midsagittal Jaw Motion and Multi-Channel Analysis for Sleep-Disordered Breathing Screening. 2008.

[2] Yoshua Bengio Geoffrey Hinton Yann LeCun. Deep learning .

[3] Sutskever I. Hinton G. Krizhevsky, A. ImageNet classification with deep convolutional neural networks. . *Proc. Advances in Neural Information Processing Systems 25 1090–1098 (2012).*

[4] Neupsy Key. Central sleep apnea and hypoventilation syndromes. `https://neupsykey.com/central-sleep-apnea-and-hypoventilation-syndromes/`, 2016. Accessed on 2020-06-04.

[5] T. Young, M. Palta, J. Dempsey, J. Skatrud, S. Weber, and S. Badr. The occurrence of sleep-disordered breathing among middle-aged adults. *New England Journal of Medicine*, 328(17):1230–1235, 1993.

[6] Christian M Langton Juha Töyräs Brett Duce, Antti Kulkas. The AASM 2012 recommended hypopnea criteria increase the incidence of obstructive sleep apnea but not the proportion of positional obstructive sleep apnea . *Sleep Medicine*, page 6, 2012.

[7] L. Parrino P. Halaász, M. Terzano and R. Bodizs. The nature of arousals in sleep. *J. Sleep Reserach*, 2004.

[8] Mary A Carskadon Paul A Easton Michael H Bonnet, David W Carley. EEG arousals: Scoring rules and examples. A preliminary report from the Sleep Disorders Atlas Task Force of the American Sleep Disorder Association . *ASDA*, page 174, 1992.

[9] R. Poirrier. Étude du comportement de la mandibule au cours des arythmies ventilatoires du sommeil: contribution à la physiopathologie du syndrome des apnées obstructives et mise au point d'un système de dépistage. 1998.

[10] Julien Simar. Machine learning techniques applied to sleep-disordered breathing diagnosis . 2020.

[11] K Hosaka K Uchida Y Yamashiro 1, Y Suganuma. Usefulness of arousal for the diagnosis of sleep breathing disorder .

[12] Sonia Ancoli-Israel José S. Loredo Joel E. Dimsdale Herbert J. Yue, Wayne Bardwell. Arousal frequency is associated with increased fatigue in obstructive sleep apnea . 2009.

[13] Nomics. MANDIBLE BEHAVIOUR DURING SLEEP-DISORDERED BREATHING: A SURROGATE FOR OESOPHAGEAL PRESSURE.

[14] Wikipedia. Supervised learning .

[15] Pierre Geurts and Louis Wehenkel. Supervised learning - ELEN062-1 class, Introduction to Machine Learning .

[16] Lior RokachOded Maimon. Decision Trees .

[17] L.Breiman. Random Forests. 2001.

[18] sklearn. RandomForestClassifier .

[19] Quoc V. Le Prajit Ramachandran, Barret Zoph. Searching for activation functions .

[20] Ilya Sutskever Alex Krizhevsky and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks .

[21] Min Marvin Wang. Rectified Linear Unit (ReLU) and Kaiming Initialization . 2019.

[22] Jimmy Lei Ba Diederik P. Kingma. Adam : a method for stochastic optimization . *ICLR 2015*, 2019.

[23] Christian Szegedy Sergey Ioffe. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015.

[24] Gilles Louppe. Convolutional networks . *INFO8010-Deep Learning*.

[25] S. Mallat. A Wavelet Tour of Signal Processing. *Academic Press*, 1999.

[26] Michael Unser Akram Aldroubi. Wavelets in Medicine and Biology. *CRC Press*, 1996.

[27] Alexander Craik et al. Deep learning for electroencephalogram (EEG) classification tasks: a review. 2019.

[28] Bharath Ramsundar Volodymyr Kuleshov Mark DePristo Katherine Chou Claire Cui Greg Corrado Sebastian Thrun Andre Esteva, Alexandre Robicquet and Jeff Dean. A guide to deep learning in healthcare . 2019.

[29] Pawel Falat Dmytro Sabodashko Veronika Herasymenko Lukasz Wieclaw, Yuriy Khoma. Biometric Identification from Raw ECG Signal Using Deep Learning Techniques. 2017.

[30] Laxmidhar Behera Chandan Kumar Behera, Tharun Kumar Reddy and Bishakh Bhattacarya. Artificial Neural Network based arousal detection from sleep electroencephalogram data .

[31] Jong-Uk Park Eun Yeon Joo Kyoung-Joung Lee Urtnasan Erdenebayar, Yoon Ji Kim. Deep learning approaches for automatic detection of sleep apnea events from an electrocardiogram . 2019.

[32] Balaji Goparaju-Brandon Westover Jimeng Sun Siddharth Biswal, Haoqi Sun and Matt T Bianchi. Expert-level sleep scoring with deep neural networks. 2018.

[33] Wikipedia. Apnea-Hypopnea Index.

[34] Takeshi Tanigawa Hiroshi Nakano, Tomokazu Furukawa. Tracheal Sound Analysis Using a Deep Neural Network to Detect Sleep Apnea .

[35] Sugata Munshi Debangshu Dey, Sayanti Chaudhuri. Obstructive sleep apnoea detection using convolutional neural network based deep learning framework. 2017.

[36] Dirk Deschrijver Tom Van Steenkiste, Willemijn Groenendaal and Tom Dhaene. Automated Sleep Apnea Detection in Raw Respiratory Signals using Long Short-Term Memory Neural Networks . *Journal of biomedical and health informatics*, 2018.

[37] Birmohan Singh Dalwinder Singh. Investigating the impact of data normalization on classification performance . 2019.

[38] PhD1 ; Samuel T. Kuna MD; Allan I. Pack-MBChB PhD ; James K. Walsh PhD ; Clete A. Kushida MD PhD ; Bethany Staley RPSGT ; Grace W. Pien MD MSCE Magdy Younes, MD. Reliability of the American Academy of Sleep Medicine Rules for Assessing Sleep Depth in Clinical Practice .

[39] MD2 ; Peter A. Cistulli MD PhD3 ; Annette C. Fedson PhD4 ; Thorarinn Gíslason MD PhD5 ; David Hillman MBBS6 ; Thomas Penzel PhD7 ; Renaud Tamisier MD PhD8 ; Sergio Tufik MD PhD9 ; Gary Phillips MAS10; Allan I. Pack MBChB PhD Ulysses J. Magalang, MD1 ; Ning-Hung Chen. Agreement in the Scoring of Respiratory Events and Sleep Among International Sleep Centers .

[40] F.A.A.S.M.1 ; Steven Van Hout B.S Richard S. Rosenberg, Ph.D. The American Academy of Sleep Medicine Inter-scorer Reliability Program: Respiratory Events .

[41] Jill Raneri RPSGT2 ; Patrick Hanly MD Magdy Younes, MD1. Staging Sleep in Polysomnograms: Analysis of Inter-Scorer Variability .

[42] Ross Girshick Kaiming He Piotr Dollár Tsung-Yi Lin, Priya Goyal. Focal Loss for Dense Object Detection . *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[43] Anca Hangan György Kovács, Gheorghe Sebestyen. Evaluation metrics for anomaly detection algorithms in time-series .

[44] Jason Brownlee. Stacking Ensemble Machine Learning With Python . *Machine learning mastery*, 2020.