

URV UNIVERSITAT ROVIRA I VIRGILI
UOC UNIVERSITAT OBERTA DE CATALUNYA

MASTER IN COMPUTATIONAL AND MATHEMATICAL
ENGINEERING

FINAL MASTER PROJECT

Soundless: A study on noise sleep disturbance

Author
Santi MARTÍNEZ PÉREZ

Supervisor
Prof. Pedro GARCÍA LÓPEZ

May 21, 2025

Contents

Introduction	1
Soundless	2
The Human Sleep Project	2
State of art	3
Implementation	4
Feature engineering	5
Channel selection	6
Classes	8
Fully Convolutional neural network (FCN)	8
LSTM	9
Band power aggregation	10
Delta activity	11
Theta activity	11
Alpha activity	11
Sigma activity	12
Beta activity	12
Power Spectral Density (PSD)	12
Multi Layer Perceptron (MLP)	13
Evaluation	14
Conclusion	16
Future work	17
Bibliography	18

Introduction

Sleep quality is an important factor for good health. Nowadays the human population is living in megalopolis, with high density population in small areas, giving rise to noisy ambients the 24 hours of the day. Studying how noise can affect sleep quality seems like a worthwhile topic to which we should devote time. The following pages presents a first approach to an sleep quality study.

Soundless

The [Soundless](#) app is a project leaded by Universitat Rovira i Virgili at Tarragona, Spain. The project aims to study how noise affects the sleep quality by providing a bracalet equiped with sensory to monitor vital signs of the subject. Along with the bracalet, an android app is provided to record the sound in the ambient while the bracelet is working. Ideally we must be able to see some arousals in the vital signs signals when high noise is recorded by the mobile application, so we can conclude how noise affects or not the sleep quality. The challenge is, however, not trivial. Since the bracelet will only record oxygen saturation and heart rate, to arrive at any conclusion is hard to do. In top of that the sleep itself is a field of study and a complex topic: there are several stages at sleep, noise may affect differently in different stages. There may be subjects more used to noise in the ambient while sleeping and noise at different frequency ranges may affect differently to different subjects.

Additionally, there are not many subjects in the sounless app study yet, around 20 people participated in the study so we looked for alternatives datasets to start our study.

The Human Sleep Project

The [Human Sleep Project](#) is a project carried out by several American hospitals, they are recording several electroencephalograms of subjects with some sleep pathology. The documents, stored in .edf (European Data Format) format are usually accompanied with

annotations on the electroencephalograms as well as the channels used for the recording. This is a vast dataset with around 25000 encephalograms to study. With this information makes sense to try to develop some kind of artificial intelligence solution to annotate encephalograms automatically, this is the main focus of this study. At some point, the technology used here may help to detect arousals related to noise in the ambient using similar methodologies.

State of art

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Implementation

In this section we will present the different approaches to the problem that have been used in the development process, some of them have been dropped for the poor results obtained, and the last one presented here is the main proposal of the document, giving fair results, and aiming to be a useful tool in electroencephalography classification. Due to the nature of the dataset, more that 25000 files, weighting around 500MB in average, the training process of neural networks presented here is not simple, we need in a first phase to download the data, in a second stage, the data must be treated and aggregated, and finally the network should be trained.

The whole [implementation](#) is coded in python, a friendly language for IA related tasks, having great libraries such as keras, tensorflow and scikit learn, that we will use extensively.

For the download process, [boto3](#) library is used to interactuate with AWS s3 instances hosting the data.

To treat the data, [MNE](#) library is used, this one comes in handy to deal with .edf files, allowing us to get the epochs of an encephalogram related to a particular annotation and computing the PSDs associated with each event.

To aggregate the data a [postgresql](#) DDBB have been used, when the data is downloaded and treated is aggregated in different batches and saved into a relational database, the neural network will read those chunks of data for the training process.

Finally to create the neural networks proposed here [keras](#) is used. This is a great library abstracting the tough part of neural networks and allowing for quick prototipe a neural network.

Feature engineering

The vast amount of data comming from encephalogram files must be treated before we can feed this data to a neural network. First of all we need to get a glimpse at the channels

used for the encephalogram recording, with some examples we realize that there are several kinds of configurations for encephalograms, a neural network won't work if we do not standardize the data between the different encephalograms. Once we select a set of channels representing the majority of our data set, we need to obtain the different epochs of the encephalograms associated to the annotations.

Channel selection

To select the representative channels of our dataset, we first traverse all our documents looking for the different channels configurations, this process will generate a CSV file with a fingerprint of the channels for each encephalogram, here a fingerprint means nothing but to concatenate all the channel names separated by a "|" character for each encephalogram. This [file](#) is the responsible for this data gathering process.

Once we have generated the CSV file with one fingerprint per row, we can start analysing the different channel configurations to get a set of desired channels to study. The process is coded in this [file](#). Roughly, this file will make an study of the frequency of each channel present in the dataset, and will try to look for equivalent channels, two channels are considered equivalent when two different configurations are equal by exchanging this channels. The results obtained in this process are as follows:

Channel	Frequency (%)
ABD	88.719512
AIRFLOW	75.304878
C3-M2	71.951220
CHEST	87.042683
E1-M2	84.451220
EKG	87.957317
HR	89.786585
IC	89.329268
LAT	82.164634
O1-M2	70.731707
RAT	79.878049
SNORE	91.920732
SaO2	90.396341

Table 1: Channel frequencies

So, demanding that our encephalograms have, at least, this channels we must get a representation of around a 71% of our dataset, which must be more than enough to train our neural network. Following with the obtained results, the meaningful equivalences found are as follows:

Channel equivalences									
C4-M1	CHIN1-CHIN2	E1-M1	E2-M1	EKG	F4-M1	O2-M2	LAT	RAT	
C4-M2	CHIN1-CHIN3	E1-M2	E2-M2	EKG-E1	F4-M2	O1-M1	LAT-E1	RAT-E1	
	CHIN2-CHIN3				F8-M1	O2-M1			
	CHIN3-CHIN1				Fp2-M1				
	CHIN3-CHIN2				Fp2-M2				
	Chin1-31								
	Chin1-Chin2								
	Chin1-Chin3								
	Chin1-P3								
	Chin1-P4								
	Chin2-Chin3								
	Chin3-Chin2								

Channel	Frequency (%)	Equivalence
ABD	88.719512	No
AIRFLOW	75.304878	No
C3-M2	71.951220	No
C4-M1	76.371951	Yes
CHEST	87.042683	No
CHIN1-CHIN2	91.310976	Yes
E1-M1	85.365854	Yes
E2-M1	87.347561	Yes
EKG	90.396341	Yes
F4-M1	81.402439	Yes
HR	89.786585	No
IC	89.329268	No
LAT	84.451220	Yes
O1-M2	70.731707	No
O2-M2	86.280488	Yes
RAT	82.164634	Yes
SNORE	91.920732	No
SaO2	90.396341	No

Table 2: Frequencies with equivalences aggregated

In table [Table 2](#) I summarize the frequencies obtained when aggregating the equivalences found by the code, after all we have 18 meaningful channels to play with, however we need to take into account that most of them are not electroencephalogram channels, so looking for brain activity on those channels will make no sense, we will review this part later on. However we can try to use the 18 channels to feed a neural network as they are, as a sequence of voltages per channel over time.

Classes

If we take a look at the annotation documents associated to each encephalogram there is a huge number of distinct annotations for the whole set, however if we look carefully, we realise that the most prevalent annotations in all the encephalograms are the sleeping stages. It makes sense to try to classify the sleeping stages of a subject, the sleeping stages are:

- **Sleep_stage_W:** This stage indicates that the subject is awake.
- **Sleep_stage_N1:** This stage indicates that the subject is in the first sleep stage.
- **Sleep_stage_N2:** This stage indicates that the subject is in the second sleep stage.
- **Sleep_stage_N3:** This stage indicates that the subject is in the third sleep stage.
- **Sleep_stage_R:** This stage indicates that the subject is in REM (Rapid Eye Movement) stage.

So these will be our class set, let's try to classify our encephalogram events into one of these stages. We can try to use a similar method to look for different kind of arousals in the encephalograms and, later on, try to associate those arousals to noise in the ambient.

Fully Convolutional neural network (FCN)

Based on [this](#) paper and the related work found [here](#) I tried to build a fully convolutional network for timeseries classification, the architecture is as follows:

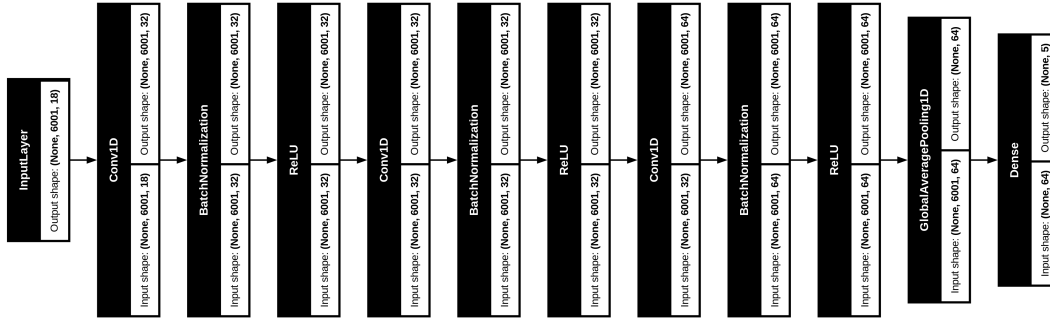


Figure 1: FCN architecture

The input shape is (6001, 18) given that we have 18 channels and that we are analysing cropped segments of the encephalogram 30 seconds long. The sample frequency of the encephalograms is 200 Hz, so we have 6000 samples per event. Any encephalogram found with a different sample frequency will be discarded in this approach. Then we have a convolutional layer of 32 filters with a kernel size of 3. After that another convolutional layer with again 32 filters and a kernel size of 5. Finally we use a last convolutional layer with 64 filters and a kernel size of 5. Then we use an average pooling layer to downsample the results of convolutional layers. Finally a dense layer with 5 neurons, the number of classes we would like to classify.

The activation function I'm using for the hidden layers of the network is [ReLU](#)(rectified linear unit) since is very efficient and mitigates the [vanishing gradient](#) problem. For the output layer softmax is used, since we have a multiclass classification problem.

The parameters of this neural network have been optimized by using [keras tuner](#) a tool to automate the process of selecting the hyperparameters of our network. It is not perfect for us since we are dealing with a huge dataset hosted elsewhere, so we need to tune the model with a few samples of the dataset, but is a helping tool anyway.

Several options have been tried for the optimizer with this architecture. In particular Adam, SGD (Stochastic gradient descent) and AdamW with a weight decay of 1e-4. At the same time several learning rate values ranging from 1e-3 to 1e-5 have been tried and a [LearningRateScheduler](#) have been applied too.

After one day of training this approach could not overcome a 20% of validation accuracy, which is not better than a random choice of one of our classes. We need to take into account that we are feeding the network with a very different selection of channels what will probably lead to a very noisy data set. After applying data scalation with MinMax scaler the accuracy improved a bit, but still not enough to make it a useful tool.

LSTM

LSTM (Long short-term memory) neural networks are a kind of recurrent neural network that are very well suited for time series classification, by using an LSTM layer instead of our 2 last convolutional layers we may get better results. The architecture proposed is as follows:

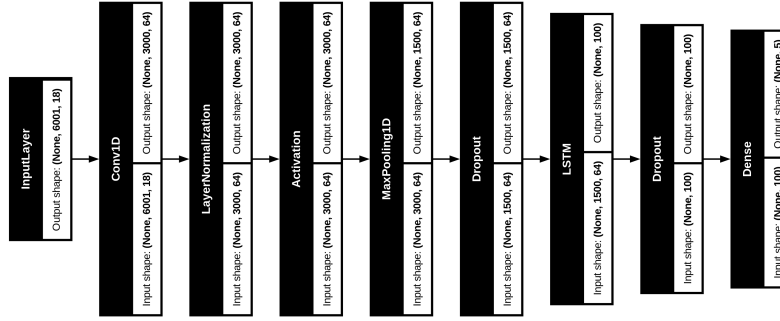


Figure 2: CN-LSTM architecture

We keep the same input layer, with shape (6001, 18). After that we use an initial convolutional layer with 64 filters and a kernel size of 3, which is good at learning general patterns in our data after that we use a normalization layer that we connect with a ReLU activation layer, finally we use a pooling layer to downsample the results obtained in this first step. After that we connect a dropout layer to avoid overfitting, this layer will "deactivate" a percentage of the neurons of the interconnected layers to trying to avoid that the weights of our gradient are too much affected by noisy data. After that we plug the LSTM layer with 100 neurons and the default tanh activation function, a dropout layer is added after this one and finally we have the output layer, with 5 neurons, again the number of classes we would like to classify.

For this architecture several optimizers has been tried too, particularly Adam with learning rates ranging from 1e-3 to 1e-5 and AdamW with a weight decay of 1e-4.

After several days training this architecture was not able to overcome 45% in validation accuracy, an improvement over fully convolutional neural networks, not enough however.

Band power aggregation

Until now I did not have too much success, a 45% on validation accuracy is still not a good result. Most probably we are dealing with very noisy data not representative of the sleep stage in the subjects, as an instance ABD or IC channels does not seem a good choice to

study the sleep stages of a subject. In top of that try to feed a neural network with the raw readings of an encephalogram seems a pretty poor choice, since there may be a lot of artifacts messing our data.

We can think in an encephalogram as a bunch of signals, but, the brain activity is characterized by particular frequencies appearing or desaparearing at different moments in the brain, if we were able to study the influence of those frequencies en each of our events we will have a much more cleaner dataset.

Delta activity

Delta brainwaves, also known as delta activity, are the slowest brainwaves, typically between 0.5 and 4.5 Hz. They are primarily associated with deep, dreamless sleep, where the body is in a state of complete relaxation and the mind is minimally active. Delta waves are also observed in newborns, and can be induced in a waking state by experienced meditators.

They usually show a high amplitude and are most prevalent during deep sleep stages (3), where the brain is essentially shut down for repair and regeneration. Delta waves are linked to various restorative processes, including tissue repair, muscle growth, and hormone release. They are also involved in transferring information from the hippocampus to the neocortex, aiding in long-term memory storage.

Theta activity

Theta brainwaves, characterized by frequencies between 4.5-8.5 Hz, are associated with states of relaxation, deep meditation, and creativity. They are often observed during sleep, particularly during dream sleep, and are also linked to states of daydreaming and focused attention. Theta activity is thought to play a role in processing information, making memories, and facilitating intuitive insights.

Disrupted or divergent theta activity has been linked to various mental health conditions.

Alpha activity

Alpha brainwaves, with a frequency of 8.5-11.5 Hz, are associated with a relaxed, awake state, often observed during activities like daydreaming or meditation. They indicate the brain is active but not overly focused, promoting feelings of calmness and ease.

Alpha activity is typically strongest over the posterior areas of the head, particularly the occipital cortex. Alpha waves can be influenced by various factors, including sensory

stimuli, mental activities, and even the position of the eyes (they are often more prominent with eyes closed).

Sigma activity

Sigma activity, also known as sigma power or spindle activity, refers to brainwave activity in the frequency range of 11.5-15.5 Hz, a key component of non-REM sleep. These brainwaves are characterized by transient oscillations and are often associated with sleep-dependent memory consolidation and sleep stability. In infants, sigma activity is linked to psychomotor development and local brain maturation.

Beta activity

Beta activity in the brain refers to a range of brainwave frequencies, typically between 15.5 and 30 Hz, associated with alert, focused, and active mental states. These brainwaves are commonly present during waking hours and when engaged in tasks requiring concentration, problem-solving, or decision-making.

An excessive beta activity can be associated with symptoms of brain over-arousal, such as anxiety, obsessiveness, and sleep difficulties, while a deficient beta activity may be linked to difficulties concentrating, problem-solving, and overall brain under-arousal.

Power Spectral Density (PSD)

Power Spectral Density (PSD) analysis is used to examine the spectral composition of EEG signals during sleep, providing insights into different sleep stages and potential sleep disorders. It's a quantitative EEG (qEEG) technique that measures the distribution of power across different frequency bands within the EEG signal.

Inspired by [this](#) article, I used the frequencies spectrum derivated from the PSD analisys of the EEGs, obtaining a vector of "features" characterizing which amount of power we find in each channel for the frequencies related to brain activity.

However, the channels we were using until the moment are not the best choice to look for brain activity, hence the list of meaningful channels we will study has been cropped into an smaller but more representative list, the channels we will use for PSD Analisis are the following:

Channel	Frequency (%)	Equivalence
C3-M2	71.951220	No
C4-M1	76.371951	Yes
E1-M1	85.365854	Yes
E2-M1	87.347561	Yes
F4-M1	81.402439	Yes
O1-M2	70.731707	No
O2-M2	86.280488	Yes

Table 3: Channels for PSD Analysis

By computing the PSD of the electroencephalograms for these channels we will obtain a vector of 35 features, the amount of power in each of the brain related activity frequencies (presented above) for each of the channels, we will discard any other frequency range.

This time we have a much cleaner and representative dataset, we must, however think in a new network architecture that fits better with our new dataset.

Multi Layer Perceptron (MLP)

A Multi Layer Perceptron ([MLP](#)) is a feed forward neural network, characterized for having an input layer, several hidden layers and an output layer. This kind of neural networks are very well suited for classifying tabulated data. After our PSD analysis we are getting a vector of fixed "features" representing the amount of power of each frequency associated with the brain activity for each of the channels we are studying, a simple MLP must do the job properly, this time the architecture used is the following:

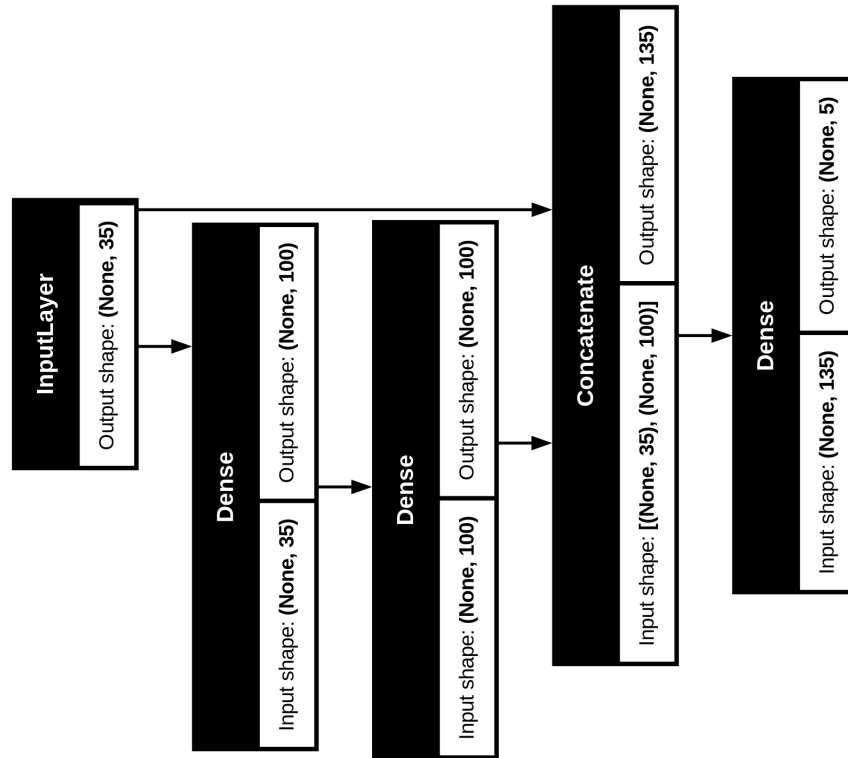


Figure 3: MLP architecture

We first have our input layer of shape (35), since we have 7 channels multiplied by the 5 frequencies we are interested in. The input is then connected to a first dense layer with 100 neurons and using the ReLU activation function, following another hidden layer of again 100 neurons and using the ReLU activation function. After the dense layers, the input is concatenated with the output of the hidden layers, so we have a deep path and a wide path for the data, this use to help the classification process when our input is very representative what we are willing to predict. Finally a dense layer with 5 neurons and an activation function of softmax is attached to the network.

The optimizer used for the MLP is Adam with an initial learning rate of $1e-3$, however a [ReduceLROnPlateau](#) is in use to reduce the learning rate when our error is not improving, allowing the network to focus more and to more stable but slow convergence.

The parameters for the number of neurons and the learning rate for this network have been optimized using [keras tuner](#).

Evaluation

At the moment of writing, the MLP has been trained with 1437 subjects, with, in average, around 850 events each patient what results in more that 1.2M events. we are reaching an accuracy of nearly a 70% in the validation phase. Now, the tests have been performed in 124 subjects taken randomly from the dataset, with about 102000 events, those subjects are nod feed into the neural network at training stage.

The [classification report](#) obtained for the predictions computed with the resulting model is the following:

	precision	recall	f1-score	support
Sleep_stage_W	0.59	0.60	0.60	13843
Sleep_stage_N1	0.33	0.26	0.29	10978
Sleep_stage_N2	0.80	0.77	0.78	47168
Sleep_stage_N3	0.70	0.73	0.71	13242
Sleep_stage_R	0.65	0.78	0.71	17240
accuracy			0.69	102471
macro avg	0.61	0.63	0.62	102471
weighted avg	0.68	0.69	0.68	102471

Table 4: Classification report

As we can see, our main problem is with class Sleep_stage_N1, where we get a precision of 33%, if we look closely into the classification report we observe that the class is the class with less support, probably applying some data augmentation techniques, like [SMOTE](#) will help the neural network to predict better for this class.

The confusion matrix obtained in the training is the following:

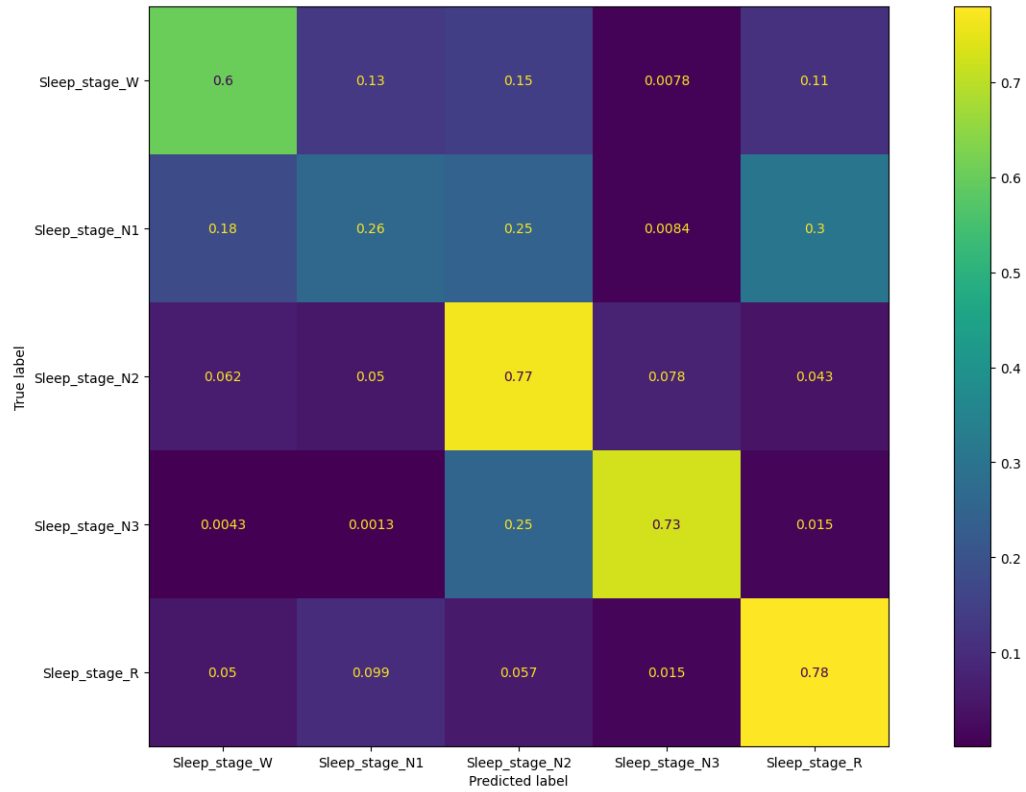


Figure 4: Confussion matrix

Here we can appreciate that our classifier is not performing well for the Sleep_stage_N1 class, being confused mainly with the class Sleep_stage_R and Sleep_stage_N2. The rest of the classes shows a better performance, being Sleep_stage_N2, Sleep_stage_N3 and Sleep_stage_R above the 70% of success in the prediction.

Conclusion

Sleep study is a complex topic, and how noise affects sleep increases the complexity of such an study. In this document however, we applied a set of techniques that allow us to predict the sleep stage of a subject given an electroencephalogram.

This techniques can be adapted or extended to study the noise disturbance on a subject while sleeping. As an instance we could categorize and classify the different arousals found in an electroencephalogram and compare the results with a recording with an standard microphone. Actually, the SNORE channel present in most of the electroencephalograms of the The Human Sleep project dataset is a microphone to capture snoring volume during sleep, however we need to take into account that this electroencephalograms has been captured in a very controled environment, so use this recordings to compareate the sounds, probably emitted by the subject itself, with arousals will probably lead to poor results, since the arousals related to snoring may not be related to noise particullarly. Ideally, with similar montages we can achieve what we want, exposing the subjects to sounds at different frequencies and amplitudes we may get a desirable result for this study. World is not ideal, neither is the data, so with the working dataset of the human sleep project this is a first step towards gaining some knowledge on how the noise affect the sleep quality.

Future work

As concluded above, the techniques used in this study may be a first step towards gaining some knowledge on how the noise affects the sleep quality. By classifying the sleep stages of a subject given an electroencephalogram we can establish in which period the noise affects more or less a subject. Besides, the generalization on sleep stage classification, it is, being able to predict the sleep stage of a subject given other subjects data is a great capability, the way we predict sleep stages we can predict arousals given another subject data. Arousal prediction will be another step forward to gain knowledge on how noise affects the sleep quality.

With a bigger dataset we could use soundless data to make a deeper study on the noise in the ambient, probably by taking the power of sound at different frequencies and trying to detect abnormalities in the signals registered by the bracelet, however, an electroencephalogram is ideal to study how sleep occurs during the resting hours.

Further steps from this study may include growing the soundless dataset, detecting arousals from vital signals and trying to establish a relationship between soundless data and electroencephalogram data. Moreover, the dataset from The Human Sleep Project provides, in some cases, a pre-sleep questionnaire. This questionnaire could be used to create a user profile, so if soundless users are willing to respond a questionnaire similar to those provided by The Human Sleep Project it will be a good point to start to relate data between the two datasets.

Bibliography

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.