Scuola universitaria professionale
della Svizzera italiana
SUPSI

University of Applied Sciences and Arts of Southern Switzerland
Department of Innovative Technologies

Applied Case Studies of Machine Learning and Deep Learning in Key Areas II

PERSONALIZED SLEEP SPINDLE DETECTION ALGORITHM

Andrea Wey
andrea.wey@student.supsi.ch

Carlo Grigioni
carlo.grigioni@student.supsi.ch

Christian Berchtold
christian.berchtold@student.supsi.ch

Professor: Francesca Faraci
SUPSI, Lugano Switzerland

15/05/2023

Table of Contents

List of Figures

1. Github repository link

Here you will find the repository of this project: click here

2. Problem definition

The aim of the project is to develop a personalized sleep spindle detection algorithm, starting from the study reported in the given paper [1].

We were asked to understand and reproduce the work presented in the above mentioned paper, and try to improve its results through critical considerations.

3. Data

We were asked to use the DREAMS dataset [2].

This Sleep Spindles Database contains 8 excerpts of 30 minutes of central EEG channel (extracted from whole-night PSG recordings), annotated independently by two experts in sleep spindles. Expert 1's scored spindle count was cut off after 1000 seconds.

We loaded the data from the txt files and resampled the signal at 200Hz to have a consistent frequency for all the data, and we apply a bandpass filter between 0.3 and 35 Hz. We then do the same with the raw excerpts and save all our data.

To extract the features of the excerpt we used a sliding window of half the sampling frequency and calculated the following metrics at each timestep: **entropy, maximum, minimum, variance, standard deviation, phase-amplitude coupling, instantaneous frequency, energy ratio, kurtosis, skewness, power peak, power ratio, interquartile range, zero crossing**.

These features are the same ones used in the paper [1] given to us for review. We extract these features for every patient and save them to a data frame. After that, we proceed by splitting the patients' data into train and test sets.

4. Data preprocessing

The first thing we did for preprocessing was to split the different patients for our train/test sets, where two out of the eight patients were put. It is very important to separate by patients and not by spindles or sequences, because that would result in data leakage, and the model would just identify correctly in the test set not because it's performing well, but because it has already seen the same patient in the training set.

We then try different balancing methods for the data, since the difference for the target class is pretty significant. First, we normalized the data, for our models to perform better, then we tried with undersampling the majority class and then oversampling the minority class. At last, we tried SMOTE to produce some synthetic data.

Our final dataset had this distribution of the labels, and this distribution:

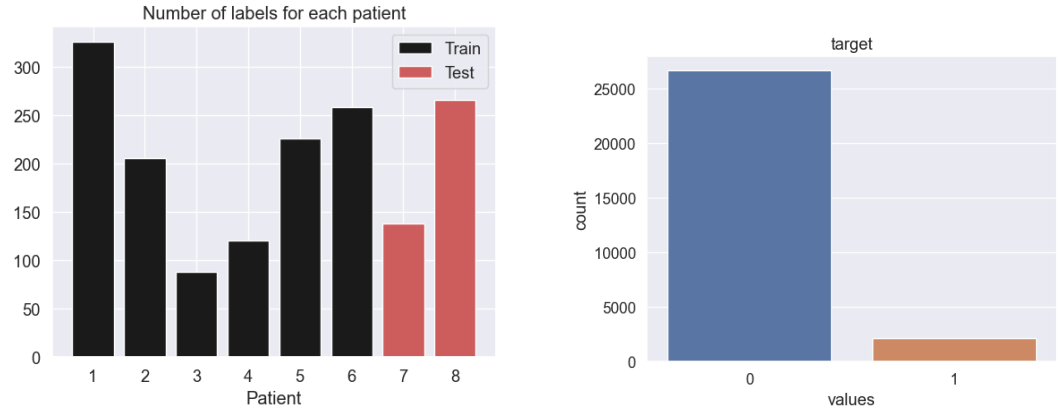


Figure 1: Label distribution

5. Paper modifications

5.1 Intuitions from the paper

When reading the paper [1] one of the things that caught our eye was the methods used for the splitting of the data. In the paper, 15 time-windows with a spindle and 15 time-windows without a spindle from the same patient were used to train the model. It was then tested on time windows from the entire signal. Given the imbalance in the dataset, the idea of undersampling the negative class to 15 samples is one of the options.

5.2 Improvements

We noticed three points where it was possible to improve modeling:

- Increase the size of the training data. 30 samples may be few to properly train complex models.
- Test different rebalancing techniques, like oversampling or SMOTE.
- Avoid any data leakage by training the model on parts of the signal which will not be part of the testing. By using time windows in the training set, which are also part of the testing set performances may appear to be better than what would happen when dealing with completely new data from new patients.

6. Modelling

We tried a couple of models to see if there was a great difference in the performances. We were surprised to see that our models did not perform as well as the one(s) in the paper.

Later in the section 7. we will list our findings for all the models used. In general, the simpler models did not seem to achieve great predictions, while for the *black box* ones, higher scores were reached. More specifically, the extreme gradient-boosting model was one of the best-performing models together with the neural network reaching a specificity and sensitivity of around 50%. We were expecting this since this is the State Of The Art (SOTA) in spindle detection are Deep Neural Networks [3].

We also tried to get a look at the F_1 score, but it always was extremely low, probably due to the great imbalance of the data. The ROC-AUC score also hovered around 50% in the best cases.

Observation: The Neural Network models were the worst performing, predicting always just the positive class when using oversampling and SMOTE.

6.1 Personalized model

XGBoost with SMOTE was trained on the signal of each individual patient, to get a more personalized model. To avoid data leakage a 0.75 split was performed on each patient.

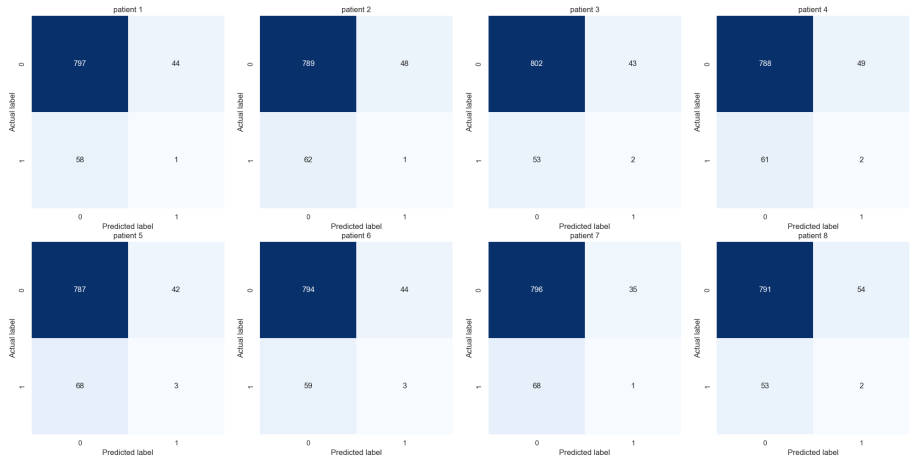


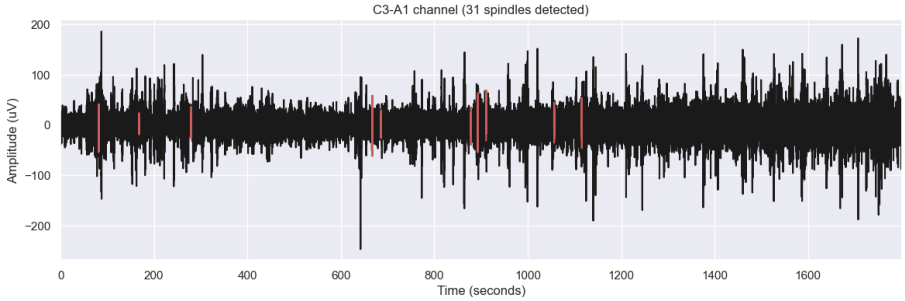
Figure 2: Results for the personalized patients

7. Results

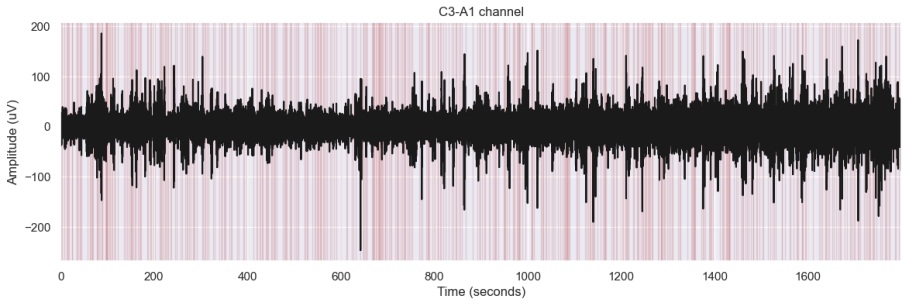
Model Results Overview						
Model Name	Undersampling		Oversampling		SMOTE	
	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
Random Forest Classifier	0.5694	0.4450	0.0000	0.9995	0.1069	0.9115
XGBoost Classifier	0.6127	0.3838	0.1503	0.8465	0.2457	0.7898
XGB Cross-val.	0.3223	0.7140	0.0670	0.9287	0.1017	0.9222
Neural Network	0.0000	1.0000	0.2269	0.8351	0.3367	0.7214

7.1 Comparing against YASA

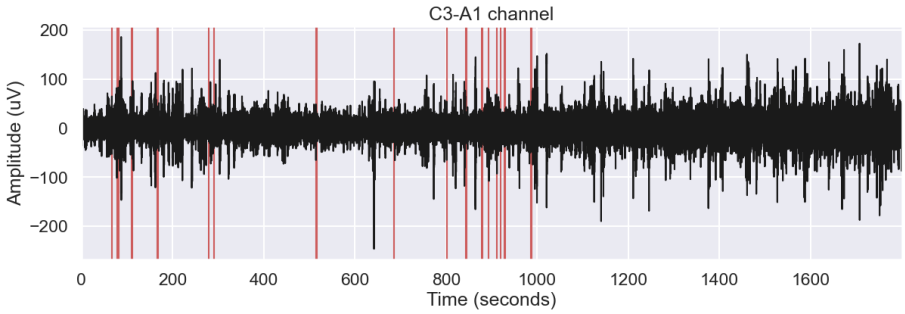
From image 3 we clearly see that our XGBoost (b) is not even close to the YASA module result (a), or the actual label (c). The model seems to find a lot more spindles (each red line is a spindle). A possible cause for this behaviour is that our model predicts the spindle as soon as there is the tiniest anomaly, while in reality, a spindle should have a higher duration.



(a) YASA spindle detection



(b) XGB predictions



(c) Expert score

Figure 3: Comparison of YASA, XGB model and expert scoring on patient 7

7.2 Non ML model

We found this article from the creator of the Python module YASA, and we wanted to test how well this method worked on our data. It turned out that the method actually seems to work well in detecting spindles.

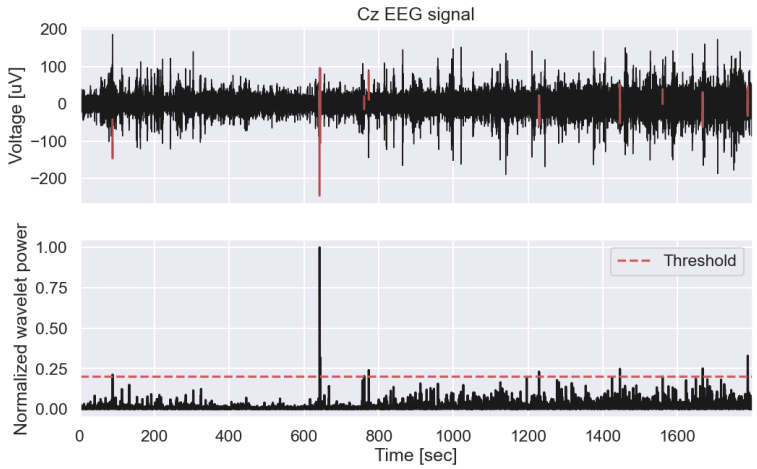


Figure 4: Morlet wavelet spindle detection method

We later discovered that in SOTA methods this wavelet function is used where they "propose a deep learning approach based on convolutional and recurrent neural networks for sleep EEG event detection called Recurrent Event Detector (RED)" [3].

8. Conclusions

This project was pretty interesting for us; working on an actual paper allowed us to analyze it, and form our opinion. It made it possible to apply the methods learned in class and to use critical thinking. The results achieved with our modifications in place are not great, to our advice the different expert labelling of the data may influence the performance of the models, since testing on a patient that has been labelled by another expert may bring some bias, and evaluating on the same patient used in training is not an option. With all these factors at play, we were not able to reach discrete and satisfiable performances. Finally, over the discussion of creating a personalized approach to the problem, we conclude that:

- There might be a general model that is trained on a variety of labelled data (definitely more than 12 patients), and then this could be applied to a single patient. It is important here to exclude the evaluated patient from the training set because in our opinion personalized should not mean overfitting our patient's data.
- In a real-world scenario there won't be an expert that manually labels one part of our data, which is then used to predict the rest of it, and even if this is the case no one can guarantee that the predictions are correct since a single labeller brings a lot of bias into our model.

References

[1] S. Scafa, L. Fiorillo, M. Lucchini, *et al.*, "Personalized sleep spindle detection in whole night polysomnography," vol. 2020, Jul. 2020, pp. 1047–1050. DOI: 10.1109/EMBC44109.2020.9176136.

[2] S. Devuyst, *The dreams databases and assessment algorithm*, Zenodo, Jan. 2005. DOI: 10.5281/zenodo.2650142. [Online]. Available: https://doi.org/10.5281/zenodo.2650142.

[3] N. I. Tapia and P. A. Estévez, *Red: Deep recurrent neural networks for sleep eeg event detection*, Oct. 2020. [Online]. Available: https://arxiv.org/abs/2005.07795.