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

PROJECT: SLEEP SPINDLE DETECTION

Manuel Ippolito
manuel.ippolito@student.supsi.ch

Fabio Loddo
fabio.loddo@student.supsi.ch

Paolo Guebeli
paolo.guebeli@student.supsi.ch

15/05/2023

Table of Contents

List of Figures

1. GitHub repository

At the following link you can check the whole project's GitHub repository:
Project's GitHub repository

2. Abstract

The project's main focus is to analyze the paper proposed by Francesca D. Faraci and colleagues[1] and try to explore how we can improve the results of their work. The goal of this project is to design a sleep spindle detection algorithm, based on the DREAMS database. From this database, we will extract and identify the best possible features, we will train multiple different models on these features and compare our results to the one achieved in the proposed paper.[1]

3. Introduction

Sleep spindles are EEG oscillations that occur during non-rapid eye movement (NREM) sleep phases and are linked to sleep-related brain plasticity. The American Academy of Sleep Medicine (AASM)[2] defines a sleep spindle as a train of distinct waves with frequency in the range of 11 to 16 Hz and a duration of 0.5 s. Spindles are a characteristic pattern of NREM2 sleep that may also be found in NREM3; their identification aids in the sleep grading system. Many spindle identification methods have been developed in recent decades and work well.

4. Data

In order to be able to compare to the given paper, we used the DREAMS database. The DREAMS database contains the recording of 8 patients, with spindles scored by two experts. The length of these EEG recordings is 30 minutes, and they are parts extracted from whole-night PSG recordings. The recordings have different sample rates, so we re-sampling all of them to 200 Hz, also the second expert only scored 6 of these recordings and the first expert only scored a part of them.

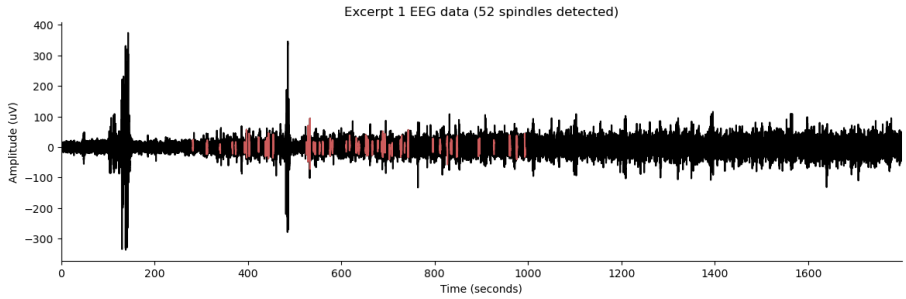

Figure 1: EEG signal for the first patient. The spindles identified with the visual analysis of the first expert are highlighted in red.

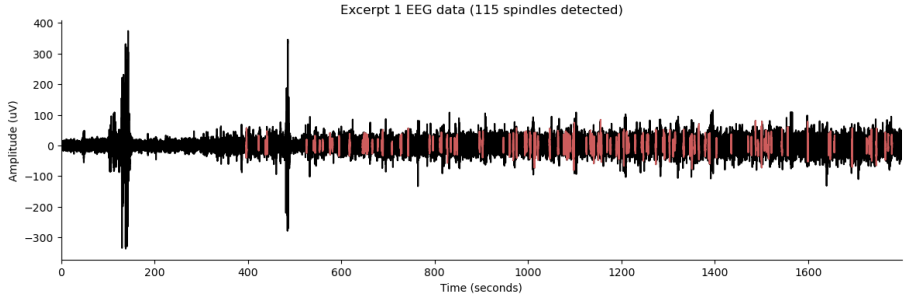

Figure 2: EEG signal for the first patient. The spindles identified with the visual analysis of the second expert are highlighted in red.

5. Feature Extraction

The first thing we had to do with the data is to find which features we are interested in and therefore extract those features. The main features we used are from the paper that was given to us [1], they are: sample entropy (SpEn), maximum, minimum, variance, standard deviation, phase-amplitude coupling (PAC) instantaneous frequency, energy ratio (Energy 11-16 Hz), kurtosis, skewness, power peak, power ratio, interquartile range (IQR) and zero crossing. To get these values we filtered the signal with a FIR filter between 0.3-35 Hz

We performed two versions of the feature extraction procedure: The first one splits the signal of each patient in windows of 0.5 seconds and then computes the features listed beforehand. In the second version, the computation was done on overlapping windows with a period of 0.5 seconds and an overlap time of 0.4. So in the end we are left with two sets of data.

5.1 Target label definition

To define the target variable we had at our disposition three main sources: The visual detection was performed by the two experts, where the second expert only scored the first six recordings, whilst the first expert only scored the first part of the 30-minute recordings. The automatic spindle detection function provided by the YASA library[3] that provide us with a simple and streamlined way approach to the task.

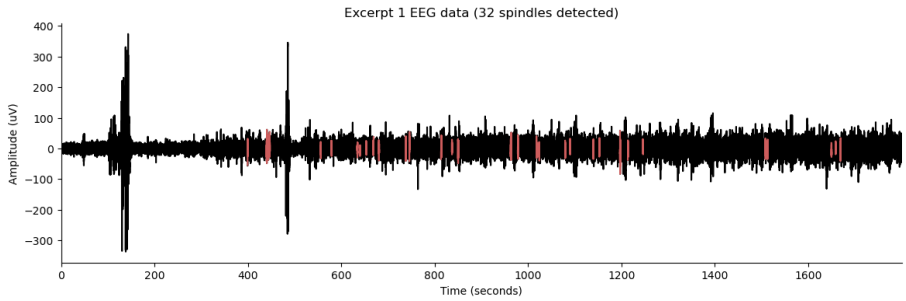

Figure 3: EEG signal for the first patient. The spindles were identified using the YASA library.

The way we constructed our label consisted basically of a union of out three sources. Since we are missing data from the 7th and 8th patient because the second expert did not perform the inspection and the YASA library method didn't find any spindles, we decided to remove them from the extraction of the features and the training/testing of the models.

6. Data augmentation

Given the very big unbalance of the target label we choose to perform some data augmentation techniques on the dataset using static windows. Our initial approach was to create a custom undersampling algorithm to balance out the two categories. We did so by removing random instances in which no spindle was detected, until we reached roughly a 50% split. A problem with using this function alone is that the actual number of spindles is very limited and we are left with a very scarce amount of data to train the model.

To fix this first problem, we thought about performing an oversampling technique on the data before balancing out the classes using undersampling. We tried a variety of different methods using a library for imbalanced data[4]:

- Random over sampler: over-sample the minority class(es) by picking samples at random with replacement.
- SMOTE: SMOTE - Synthetic Minority Over-sampling Technique as presented in [5].
- ADASYN: Oversample using Adaptive Synthetic (ADASYN) algorithm. This method is similar to SMOTE but it generates a different number of samples depending on an estimate of the local distribution of the class to be oversampled.
- Borderline SMOTE: a variant of the original SMOTE algorithm proposed in [6]. Borderline samples will be detected and used to generate new synthetic samples.
- SMOTE ENN: Combine over- and under-sampling using SMOTE and Edited Nearest Neighbours.

Taking inspiration from [1], we also made a dataset with overlapping windows of 0.5 seconds, with 0.4 seconds of overlap. This can also be seen as a data augmentation technique, as we are generating many more samples than in the original dataset.

7. Model choice

Regarding the choice of models we first tried to replicate the procedures used in the original paper by using the set of features created with overlapping windows and using a Support Vector Machine classifier, and then applying a post-processing function. This post-processing function consists of removing the lonely spindles (spindles detected without any neighbours) and grouping back the overlapping windows by a majority vote (we have 5 overlapping windows if 3 of them are spindles then all of them are.

Besides the proposed SVC model, we have also tested an XGBoost Classifier. We ended up using this model because we are interested in having models that can be explained to doctors in their functioning, and from which we can get outputs that have a basis that the doctors can understand. For these reasons, models like SVC and tree-based models seem to be a good choice. XGBoost Classifier and SVC are both models that in competitions can often reach good results while remaining quite simple to be explained.

We didn't test simpler models like K-nearest neighbours and Naive Bayes because for this task they are not really suited, so we just excluded them from our possible choices. Further implementations can be done in that way.

We have performed hyper-parameters tuning on both the selected models and for both the proposed datasets. The tuning procedure was performed using the validation set, and we opted to maximise the ROC-AUC score as it is one of the best metrics for binary classifiers.

- For SVC we used GridSearchCV and RandomizedSearchCV to tune its: linearity coefficient (C), gamma, kernel type, and degree. We tested both methods for hyper-parameter tuning and ended up using RandomizedSearchCV as it was much faster and enough precise in the results.

- For XGBoost Classifier we used the library HyperOpt, a well-known framework for hyper-parameters tuning, which unfortunately doesn't work well on SVC. The tuned parameters of XGBoost Classifier are: number of estimators, maximum depth of the tree, learning rate, minimum child nodes weight, gamma, subsample, column sample by tree, alpha regularization, and lambda regularization.

8. Results and discussion

The testing procedure was performed in two different ways:

- Following the procedure proposed in [1].
- Splitting from the beginning the patients in training, validation and testing sets. The patients were divided into training, validation and testing sets in order to reach nearly a 50% of spindles in the training set, and nearly 20% spindles in the validation set, and nearly 30% spindles in the testing set.

  To reach this balance, patients 1 and 6 were put in the training set, patients 2 and 4 were put in the validation set, and patients 3 and 5 were put in the testing set.

  As described in section 5, Data Augmentation was performed on the data to increase by about 100% the number of spindles and then balance to 50% spindles and 50% non-spindles samples in the training set. The testing set was not altered.

  No post-processing procedure was used in this case.

The results obtained can be seen in the following results table:

Model	**Sensitivity**	**Specificity**	**FPR**	**F1 Score**	**ROC AUC**	**Accuracy**
XGBC with overlaps	0.216895	0.97929	0.0207101	0.282318	0.598092	0.932898
SVC using Borderline SMOTE	0.313099	0.833164	0.166836	0.125641	0.573131	0.810556
XGBC with overlaps and Borderline SMOTE	0.495146	0.758915	0.241085	0.181172	0.627031	0.743818

Table 1: Machine Learning Model Results

The results are not impressive, but with our procedure, we are sure of not having any data leakage during the whole process, so the results are similar to what our models would obtain in a real-world environment with new unseen data.

9. Conclusions

The shortfall in appropriate data was the main cause of the not outstanding results of our models, therefore having access to larger datasets could result in big improvements in the performances.

The proposed solution remains valid and could be further explored in future research work having more data at their disposal. We have explored data augmentation techniques in the field of bio-medical signals, which is also an active field of research, and which could also bring improvements to the proposed models' performances. Performing data augmentation, both using methods such as SMOTE and also by creating overlapping windows, lead to slightly better performances of the models.

The final models' performances are not great, but as seen from the visual labelling made by doctors, there is a clear difficulty in labelling spindles also from experts of the field, so it seems to be a quite complex task. We compared the spindles detected by the doctors and the ones detected by the python library YASA [3], and there are not many detected spindles in common.

About the personalized spindle detection approach, we reached the conclusion that it is not possible to explore such an approach having this small amount of data. Also, performing personalized detection requires a doctor visually detect some spindles manually before having the possibility to build a model on them, so it is not really something applicable in the real world.

References

[1] S. Scafa, L. Fiorillo, M. Lucchini, *et al.*, "Personalized sleep spindle detection in whole night polysomnography," version latest,

[2] [Online]. Available: https://aasm.org.

[3] R. Vallat and M. P. Walker, "An open-source, high-performance tool for automated sleep staging," *eLife*, vol. 10, A. Peyrache, C. Büchel, and S. Bagur, Eds., e70092, Oct. 2021, ISSN: 2050-084X. DOI: 10.7554/eLife.70092. [Online]. Available: https://doi.org/10.7554/eLife.70092.

[4] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002. DOI: 10.1613/jair.953. [Online]. Available: https://doi.org/10.1613%2Fjair.953.

[6] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 878–887, ISBN: 978-3-540-31902-3.