Manuel Acquistapace
Stefano Billeter

Sleep spindle detection algorithm

Personalized version

Manuel Acquistapace
Stefano Billeter

Table of contents

Introduction

Sleep spindles, characterized by their distinctive waveform patterns on electroencephalography (EEG), are transitory rushes of brain activity typically occurring during non-rapid eye movement (NREM) sleep. Detecting these spindles accurately and efficiently is crucial in understanding sleep physiology and diagnosing sleep disorders. Digital signal processing techniques and machine learning algorithms have led to the development of sophisticated sleep spindle detection algorithms. These algorithms play a vital role in both research and clinical settings by enabling automated analysis of sleep EEG data, hence facilitating the identification of sleep stages and abnormalities.

The link to the GitHub repository is here.

Audience

The audience for our report includes:

a. Researchers in the field of sleep medicine and neuroscience who are interested in understanding the dynamics of sleep spindles and their implications for brain function and health.
b. Clinicians and healthcare professionals involved in sleep disorder diagnosis and treatment who rely on accurate analysis of sleep EEG data for patient care.
c. Developers and engineers working on designing software and hardware solutions for sleep monitoring and analysis, including wearable devices and medical diagnostic systems.
d. Students and academics studying biomedical engineering, signal processing, or related disciplines who seek to gain insights into advanced techniques for analyzing physiological data.

Problem statement

The goal of this project is to develop a personalized version of a sleep spindle detection algorithm based on the given research paper [1]. The task is to reproduce the presented work and try to improve its result through critical considerations. This report is part of the requirements for the final project of the "Applied Case Studies of ML and DL in Key Areas II" course, SUPSI AY 2023-24, BSc Data Science and AI.

Dataset

We were tasked with utilizing the DREAMS dataset [2], which comprises eight excerpts of 30-minute central EEG channel recordings (extracted from entire-night PSG recordings). These excerpts were independently annotated by two sleep spindle experts. The spindle count for Expert 1 was truncated after 1000 seconds. Then we imported the data from the provided text files and standardized the signal frequency to 200Hz through resampling. Furthermore we applied a bandpass filter ranging from 0.3 to 35 Hz to enhance signal quality.
To extract features from the excerpts, we employed a sliding window approach with a window size equal to half the sampling frequency. At each time step, we computed various metrics, including mean, standard deviation, skewness, kurtosis, zero crossing, any many more. These features align with those utilized during the course.
Following feature extraction, we organized the data into a structured format, saving it into a data frame. We partitioned then patients' data into training and testing sets to allow robust model training and evaluation.

Preprocessing

As said above, we proceeded with splitting the different patients in train/test sets, where 1 patient was used for test. In this kind of tasks it is crucial to correctly separate with respect to patients and not to

spindles, because, as seen during the course, through this kind of bio-signals contain unique information that could lead to identify the patient. This would propagate and result in data leakage and the model would then perform well because it has already seen the same patient in the training set.
We also implemented over and undersampling using the 'RandomOverSamper' and 'RandomUndersampler' from the imbalanced-learn library, this is particularly useful in machine learning applications where the number of observations in one class significantly outnumbers the observations in another, which can negatively affect the performance of a model.
Since, as said before, the dataset where manually annotated by two experts, we kept this dichotomy in our analysis. The part of the dataset annotated by expert 1 (in the notebook: Visual 1) comprehends all the 8 patients, where the first is our test. Concerning the part of the dataset manually annotated by the second expert, we took into consideration the first 6 patients, where the first is our test.
Furthermore, it is important to clarify that, for the first dataset part, we retained only the annotated part, while discarding the one without ground truth.

Research paper analysis

In the analysis of the paper [1] several critical insights and potential improvements were identified, which could enhance the methodology and outcomes of sleep spindle detection in EEG data.

1. Training Data Size
The researchers utilizes a balanced dataset comprising 15 time-windows with spindles and 15 without, from the same patient for training the model. This approach, while effective in maintaining a balance between the positive and negative classes, presents limitations due to the small size of the training set. Expanding the number of samples could potentially improve the model's ability to capture a more comprehensive range of data variabilities and enhance its generalization capabilities. A larger dataset could improve the model's robustness and accuracy.

2. Rebalancing Techniques
The study mentions a form of balancing by equally selecting time-windows with and without spindles. Exploring advanced rebalancing techniques, such as Synthetic Minority Over-sampling Technique (SMOTE) or random oversampling, could provide benefits. These methods might help in better managing the imbalanced nature of the full dataset, which could lead to enhancements in model performance, particularly in sensitivity and specificity metrics.

3. Data Leakage Prevention
A significant observation from the paper [1] is the absence of a detailed discussion on preventing data leakage between the training and testing sets. Ensuring that the training and testing data are completely independent is crucial for validating the model's effectiveness in real-world scenarios. Data leakage can lead to overfitting, where a model performs well on test data due to indirect or direct exposure to those data during training. Ensuring strict separation can help in achieving a more accurate assessment of the model's true predictive capabilities. The potential for data leakage in this research primarily stems from the methodology described for selecting training and testing data, particularly from how the time-windows are utilized within the same patient data for both training and testing the model.

Methodology and modelling

In this section we detail the methodology employed to evaluate the performance of sleep spindle detection models using datasets annotated by two different experts. Our approach integrates comprehensive data handling practices, model training, and evaluation strategies to ensure the robustness and accuracy of the models tested.

The methodology begins with the proper segregation of the data into training and testing sets. This strategy minimizes the risk of data leakage and helps in assessing the generalizability of the models across different patients.
Given the known issue of class imbalance, data resampling techniques are applied. Both undersampling and oversampling methods are employed to balance the classes in the training data. This is important to prevent the models from being biased towards the majority class and to improve their sensitivity in detecting the minority class, which in this case are the sleep spindles.

We trained several models using the all the datasets to detect sleep spindles effectively. These include Random Forest, K-Nearest Neighbors (KNN), and XGBoost, each known for their distinct advantages in handling classification tasks. Random Forest is utilized for its robustness and ability to handle overfitting, KNN for its simplicity and efficacy in capturing the proximity of data points, and XGBoost for its performance in dealing with structured data and its efficiency in large datasets.

Our models are evaluated based on several metrics including balanced accuracy (B.A.), precision, recall, and the F1 score. Confusion matrices are generated and displayed for each model to visualize the true versus predicted classifications, offering insights into the models' performance at a granular level.

Results

In this chapter, we will delve into the presentation of our results, compared to the paper's ones.
As aforementioned we are keeping the distinction between Visual 1 and Visual 2 for all the models.

Visual 1:

	Imbalanced				**Undersampling**				**Oversampling**			
	B.A.	Prec.	Rec.	F1	B.A.	Prec.	Rec.	F1	B.A.	Prec.	Rec.	F1
RF	0.8796	0.3869	0.8173	0.5252	0.9005	0.2477	0.9274	0.3909	0.8864	0.3538	0.8407	0.5010
KNN	0.5484	0.7368	0.0984	0.1736	0.6354	0.0818	0.5457	0.1422	0.6952	0.1213	0.5785	0.2006
XGB	0.8940	0.2858	0.8876	0.4324	0.8923	0.2415	0.9133	0.3820	0.8854	0.2891	0.8665	0.4335

Visual 2:

	Imbalanced				**Undersampling**				**Oversampling**			
	B.A.	Prec.	Rec.	F1	B.A.	Prec.	Rec.	F1	B.A.	Prec.	Rec.	F1
RF	0.8313	0.3600	0.7544	0.4874	0.8363	0.2536	0.8421	0.3898	0.8340	0.3273	0.7772	0.4606
KNN	0.5008	0.6667	0.0018	0.0035	0.6499	0.1171	0.6193	0.1970	0.6469	0.1475	0.4860	0.2263
XGB	0.8319	0.2835	0.8026	0.4191	0.8299	0.2444	0.8368	0.3783	0.8303	0.2818	0.8000	0.4168

Overview: The training and testing performance metrics for Random Forest, KNN, and XGBoost across the 'Visual 1' and 'Visual 2' datasets show notable trends in model behavior, overfitting issues, and effectiveness in generalization. Here, we delve into these insights, focusing on how each model copes with original, undersampled, and oversampled data scenarios.

'Visual 1' Dataset Analysis:

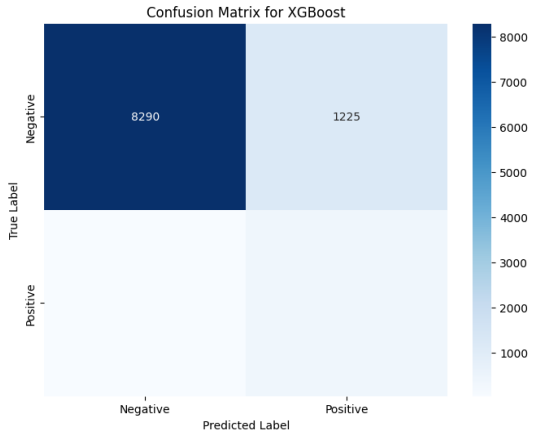
- **Random Forest** shows strong recall rates across all data scenarios, indicative of good sensitivity but at the cost of precision, particularly evident in the original and oversampled data. This trade-off leads to moderate F1 scores, suggesting a potential overfitting to the majority class while failing to adequately generalize the minority class distinctions in unseen data.
- **KNN** displays perfect training metrics, which starkly contrasts with its poor test performance, particularly highlighted by the F1 score and recall in the test phase. This discrepancy is a clear sign of overfitting, where the model memorizes the training data but fails to adapt to new, unseen data.
- **XGBoost** demonstrates robustness in handling recall across training scenarios, with a noticeable drop in precision in the test results. Although it maintains higher balanced accuracy compared to other models, the precision-recall trade-off is still apparent, affecting the overall F1 scores.

'Visual 2' Dataset Analysis:

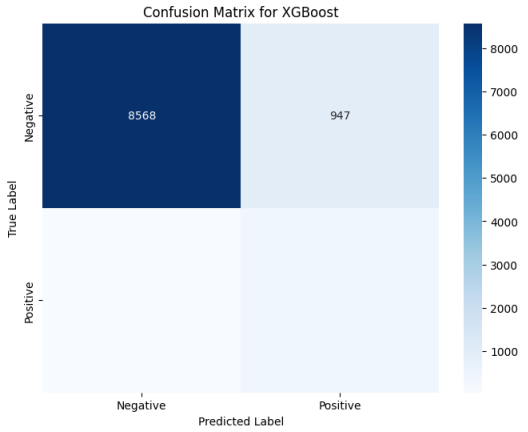
- **Random Forest** again shows consistent training performance with a high recall, indicating a tendency to overfit with similar precision-recall imbalances as seen in 'Visual 1'. Test performance shows slight improvement in balanced accuracy but continues to suffer from low precision.
- **KNN's** training results replicate the perfect scores observed in 'Visual 1', which does not translate into effective testing outcomes, demonstrating extreme overfitting.
- **XGBoost** maintains strong recall in both training and testing, though it experiences a drop in precision across test scenarios. This model offers the best balanced accuracy among the three models in testing, suggesting better generalization capabilities despite some level of overfitting.

Model Selection and Recommendations:

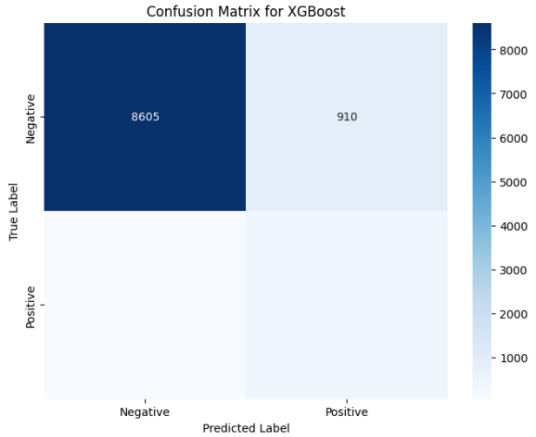
Given the detailed performance analysis, **XGBoost emerges as the preferable model** for both datasets despite its issues with precision, as it consistently shows better generalization in balanced accuracy and recall than its counterparts.



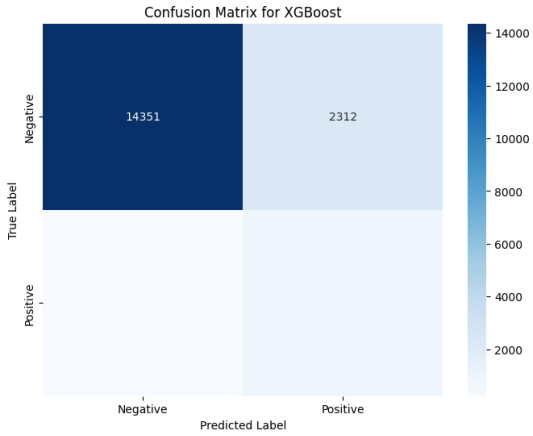
Visual 1 - original



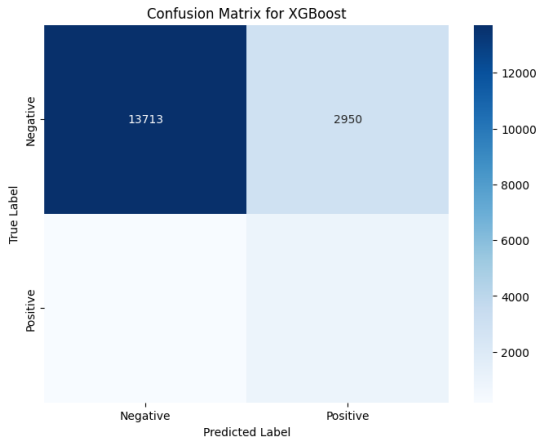
Visual 1 - undersampled



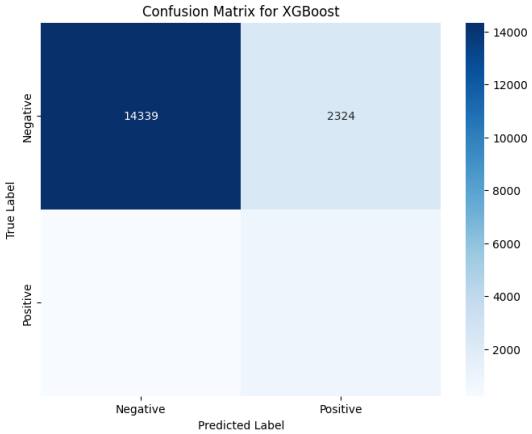
Visual 1 – oversampled



Visual 2 - original



Visual 2 – undersampled



Visual 2 - oversampled

Next steps

In this chapter, we aim to give a quick glance at some possible improvements and next steps that could be taken in order to get better and more robust outcomes.

- **Explainability**
  Evaluating the features and understand how each of them contributes to the outcomes could be crucial to improving the transparency and trustworthiness of the models. By developing a deeper understanding of the feature influences and their interactions within the models, we can not only enhance the model's performance but also ensure that it makes decisions based on clinically relevant and interpretable factors. This approach will facilitate more informed decision-making in clinical settings and improve the acceptance of automated systems by healthcare professionals.
- **Fine-tuning**
  Fine-tuning model parameters is essential to optimize performance, as it allows for the precise adjustment of learning rates, regularization strengths, and other model-specific settings that can significantly affect outcomes.
- **Further feature generation**
  Investigating the development of new features from existing data can provide deeper insights and uncover hidden patterns that are not immediately apparent, potentially increasing the accuracy and efficiency of our models.

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.