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

Dyuman Bulloni
dyuman.bulloni@student.supsi.ch

Amos Colombo
amos.colombo@student.supsi.ch

Christian Pala
christian.pala@student.supsi.ch

Professor: Francesca D. Faraci
SUPSI, Lugano Switzerland

15/05/2023

Table of Contents

List of Tables

1. Introduction

Sleep spindles are an important electroencephalography (EEG) pattern observed during non-rapid-eye-movement (NREM) sleep stages, with a frequency range of 11 to 16 Hz and a duration of at least 0.5 seconds[1]. They are considered to play a crucial role in sleep-related cerebral plasticity and are believed to mediate many sleep-related functions, from memory consolidation to cortical development [2]. Automated sleep spindle detection algorithms have been developed and show good performance, but their performance deteriorates when applied to different datasets [3].

The automated spindle detection procedure typically involves pre-processing, feature extraction, feature selection, and spindle recognition with a classifier. We follow the approach from [4], to explore if a more personalized algorithm may be beneficial for spindle detection.

In this paper [5] we try to build upon the previous work done in [4] and test various changes to update and validate their results.

2. Data

The data for this study was sourced from the DREAMS Sleep Spindles Database, a widely recognized and publicly accessible resource for sleep studies. This is the same dataset utilized in the previous study [4]. We followed identical signal processing steps as those outlined in [4]. In terms of label generation, we adopted a comprehensive approach. Initially, we employed the scoring provided with the database, as in [4]. Subsequently, we supplemented this procedure by generating new labels using the Yet Another Spindle Algorithm (YASA) [6]. Finally, we combined these labels to create a more robust and comprehensive set for analysis.

2.1 Critique of the Data

The dataset employed in the original paper, while considered standard in the research community and selected for benchmarking purposes, presents several limitations upon further scrutiny. A notable concern is the limited number of patients included in the dataset. This restricts the extent to which meaningful conclusions can be drawn regarding model performance.

Furthermore, the evaluation of this dataset was partially carried out by only two experts, covering different patients and time windows [4].

In contrast, YASA was built on a more diverse dataset evaluated by five different experts [6], achieving state-of-the-art performance in spindle detection.

We advocate for new studies on personalized spindle detection using more robust datasets.

3. Feature extraction

The codebase for the feature extraction and for all other parts of the proejct, is publicly available on our Github repository [5]. For feature extraction, we utilized the identical set of features as described in the reference [4], employing the same windowed methodology. Furthermore, we incorporated additional features from YASA [6].

- Hjorth parameters of mobility and complexity: The Hjorth parameters of mobility and complexity are measures of the temporal dynamics of the signal. The mobility parameter is a measure of the rate at which the amplitude of the signal changes over time, while the complexity parameter is a measure of the irregularity or complexity of the signal. In the context of EEG signals, the Hjorth parameters can be used as indicators of the dynamics of the underlying neural activity.

- hypnogram: A feature that provides a detailed representation of an individual's current sleeping phase, helping to determine whether it corresponds to a spindle or not.

These parameters have been found useful in characterizing the spectral properties of sleep spindles, which are known to be associated with cognitive processes such as memory consolidation and learning [6].

3.1 Feature ranking

We verified the results of [4] by ranking the features using Maximum Relevance Minimum Redundancy feature selection (MRMRF). The feature importance we derive is similar but not identical to [4], as shown in Table 1. We also integrated the ranking using Mutual Information (MI) as a different criterion. Our findings suggest that the addition of Hjorth mobility and complexity as features are valuable for the analysis.

Table 1: Feature rankings with MI and MRMRF

MI Ranking		**MRMRF Ranking**	
Rank	**Feature**	**Rank**	**Feature**
1	Phase-amplitude coupling	1	Complexity
2	Power peak	2	Sample entropy
3	Energy ratio	3	Variance
4	Power ratio	4	Power ratio
5	Complexity	5	Hypnogram
6	Zero-crossing rate	6	Mobility
7	Hypnogram	7	Zero-crossing rate
8	Mobility	8	Inter-quartile range
9	Sample entropy	9	Power peak
10	Maximum value	10	Skewness
11	Kurtosis	11	Kurtosis
12	Mean frequency	12	Energy ratio
13	Inter-quartile range	13	Mean frequency
14	Standard deviation	14	Phase-amplitude coupling
15	Variance	15	Standard deviation
16	Minimum value	16	Minimum value
17	Skewness	17	Maximum value

Feature selection often comes with the risk of information loss. Although it can improve model interpretability and computational efficiency, it may inadvertently remove relevant information, which could negatively impact model performance. Considering the complexity and multifaceted nature of sleep spindle detection, it was deemed more prudent to retain the full feature set to encapsulate as many aspects of the data as possible.

This was also motivated by the differences in the rankings produced by the MRMRF and MI methods, as shown in Table 1, which indicated that there is no clear consensus on the most important features. Consequently, the elimination of any feature could potentially result in the loss of valuable information, justifying our decision to retain the complete feature set in this study.

4. Label generation

We used spindle detection annotations provided by two experts and an automatic algorithm, as described in [4]. Following the approach outlined in [4], we created our ground truth by combining the detected spindles from all sources. Additionally, we included spindles detected by the YASA algorithm. Our labeling approach considered a window as containing a spindle if any of the labeling methods identified a positive value within that time interval. We followed the procedure in [4] to remove outliers, where a window with a positive spindle detection surrounded by windows of negative detection is considered as such. This labeling methodology was adopted to address the scarcity of positive samples in the dataset, assuming that a sample was a spindle if at least one source labeled it as such. In scenarios with more data availability, a more rigorous methodology would be preferred to consolidate the various sources.

5. Modeling

In the original paper [4], the authors employed a Support Vector Classifier (SVC) for their analysis. In our study, we expanded upon this by not only implementing the SVC but also exploring additional machine-learning models. The models chosen for this study include the SVC, K-Nearest Neighbors (KNN), Random Forest (RF), and Gradient Boosting (GB). Briefly, these models are characterized as follows:

- **SVC**: A versatile algorithm widely used for its efficacy in classification tasks with clear class separations.
- **KNN**: An intuitive model that performs well in datasets with fewer features, particularly when data points cluster within classes.
- **RF**: An ensemble algorithm adept at handling high-dimensional and noisy datasets, offering robustness against overfitting.
- **GB**: An ensemble model known for its effectiveness in binary classification tasks and capability to handle complex datasets with numerous features.

By applying these diverse models, our aim is to explore a range of methodologies and assess their applicability to our sleep spindle detection task.

5.1 Training, Validation, and Testing

In the referenced paper [4], the authors detail a training and testing procedure to compare personalized and general models. We identified several issues with this approach, particularly:

1. The choice of a very small dataset, 30 observations, for both the global model and individual patients.
2. The random selection of 15 positive and 15 negative instances potentially introduces selection bias and leads to opaque results.
3. Testing on the entire signal after sampling instances for training can result in overoptimistic performance estimates.

We used a conventional approach for data partitioning to address these concerns. For individual models, we split the data into 80% for training and 20% for testing. For the global model, we randomly selected 5 patients for training, 1 for validation, and 2 for testing. This approach prevents data leakage in the global approach. The DREAMS database only contained patients with sleep anomalies, lacking a "normal" baseline, which makes the task hard.

5.1.1 Normalization

We employed the Standard Scaler from Scikit-learn [7] to normalize the data. This normalization process was integrated into a single pipeline for each model to streamline the procedure and ensure consistency across different models.

5.1.2 Evaluation Metrics

Given the significant class imbalance inherent to our dataset, and our prioritization of the positive class (spindle), we adopted two distinct metrics for assessing our models:

- **Binary F-1 Score**: This metric computes the F-1 score solely for the positive class. It is particularly valuable in our case as we are primarily concerned with correctly identifying sleep spindles.
- **Macro F-1 Score**: This metric calculates the unweighted average of the F-1 scores for each class. It provides a holistic measure of the model's performance across all classes without considering class unbalances.

5.2 Data Augmentation

Given the scarcity of positive samples in the database, the preliminary unbalanced models struggled to generalize and often disregarded the minority class. To rectify this, we experimented with various strategies using the imbalanced-learn [8] library:

- **Random Under-Sampling (RUS):** Randomly eliminates samples from the majority class to harmonize the class distribution. Although straightforward, it risks discarding potentially important samples, leading to loss of information.
- **Random Over-Sampling (ROS):** Duplicates random samples from the minority class to balance the class distribution. While beneficial when the minority class has relatively few samples, it can lead to overfitting.
- **SMOTE (Synthetic Minority Over-sampling Technique):** Generates synthetic samples for the minority class based on existing samples' interpolation. It can be effective when the minority class is underrepresented and aids in preventing overfitting.
- **SVM SMOTE:** A variant of SMOTE that employs a Support Vector Machine (SVM) to identify the optimal samples for generating synthetic instances. It can outperform conventional SMOTE when the minority class significantly overlaps with the majority class.
- **ADASYN (Adaptive Synthetic Sampling):** Produces more synthetic samples for the minority class near the decision boundary and fewer samples farther from the boundary. It can be beneficial when the decision boundary between classes is complex and challenging to model.

5.3 Post processing

Adhering to the methodology outlined in [4], we implemented a corrective step to handle potential outliers predicted by the models. Specifically, we ensured that no spindle window was predicted in isolation.

5.4 Global models comparison

Table 2 presents the performance metrics of various algorithms on the global dataset. Mirroring the findings of [4], our models also achieve the best spindle detection results with the Support Vector Classifier (SVC). However, contrary to the results reported in the referenced paper, our scores significantly lag behind the current state-of-the-art benchmarks.

Unbalanced Dataset		
Model	**Macro F-1**	**Binary F-1**
SVC	0.577	0.181
KNN	0.579	0.187
RF	0.588	0.202
GB	0.612	**0**.250
Balanced Dataset SVC		
Sampler	**Macro F-1**	**Binary F-1**
RUS	0.594	**0.291**
ROS	0.581	0.279
Balanced Dataset GB		
Sampler	**Macro F-1**	**Binary F-1**
RUS	0.552	0.212
ROS	0.560	0.211

Table 2: Global model comparison table

6. Hyper-parameter tuning

Hyper-parameter tuning was not mentioned in the original paper[4]. We used the Optuna framework [9] to improve the scores obtained by our baseline models. For some patients, and on the global model, an improvement of 10-45% on the baseline binary F1-score was achieved.

7. Results

To compare the global and personalized approaches, we report our findings using both methodologies.

7.1 Personalized models

The best combination of models and samplers for each patient is shown in Table On average we

Table 3: Best sampler and model combination for each patient

Patient	Sampler	Model	Macro F-1	Binary F-1
1	ADASYN	SVC	0.699	**0.458**
2	ADASYN	GB	0.589	0.317
3	SMOTE	SVC	0.676	0.372
4	SMOTE	SVC	0.698	0.424
5	RUS	RF	0.513	0.361
6	ADASYN	KNN	0.615	0.301
7	SVM SMOTE	KNN	0.492	0.187
8	RUS	GB	0.658	0.365

obtained an F1 binary score of **0.348** for the personalized models.

7.2 Global models

In Table 4 we report the results of the global SVC model with the original features from [4] and the new set we developed for this paper.

Features	Macro F-1	Binary F-1
original	0.513	0.053
augmented	0.577	0.181
fine-tuned	**0.629**	**0.322**

Table 4: Global SVC results

7.3 Discussion

The personalized approach did not show a significant improvement compared to the global model. The variability in the results highlights the need for a more thorough analysis. However, there are promising indications, especially in the case of Patient 1 for which we had more detected spindles, where we achieved a substantial improvement in detection compared to the global baseline.

8. Conclusions

Our study indicates that while personalized models hold promise, they do not necessarily outperform global models in spindle detection tasks. This underscores the importance of a balanced approach, leveraging both global and individual patient data as well as different classification strategies.

References

[1] S. Foundation, *Sleep spindles*, https://www.sleepfoundation.org/how-sleep-works/sleep-spindles, Accessed on May 14th, 2023.

[2] T. Andrillon, Y. Nir, R. J. Staba, *et al.*, "Sleep spindles in humans: Insights from intracranial eeg and unit recordings," *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 31, no. 49, pp. 17 821–17 834, 2011. DOI: 10.1523/JNEUROSCI.2604-11.2011. [Online]. Available: https://doi.org/10.1523/JNEUROSCI.2604-11.2011.

[3] S. C. Warby, S. L. Wendt, P. Welinder, *et al.*, "Sleep-spindle detection: Crowdsourcing and evaluating performance of experts, non-experts and automated methods," *Nature Methods*, vol. 11, no. 4, pp. 385–392, 2014. DOI: 10.1038/nmeth.2855. [Online]. Available: https://doi.org/10.1038/nmeth.2855.

[4] S. Scafa, L. Fiorillo, M. Lucchini, *et al.*, "Personalized sleep spindle detection in whole night polysomnography," *Frontiers in Neuroscience*, vol. 15, p. 638 232, Jul. 2020. DOI: 10.3389/fnins.2021.638232.

[5] D. Bulloni, A. Colombo, and C. Pala, *Github project repository*, https://github.com/ChristianPala/spindle_detection, Accessed on May 15th, 2023.

[6] R. Vallat and M. P. Walker, "A universal, open-source, high-performance tool for automated sleep staging," *bioRxiv*, 2021. DOI: 10.1101/2021.05.28.446165. eprint: https://www.biorxiv.org/content/early/2021/05/28/2021.05.28.446165.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2021/05/28/2021.05.28.446165.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[8] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, *et al.*, "Personalized spindle detection-Python," *PeerJ*, vol. 2, e453, Jun. 2014, ISSN: 2167-8359. DOI: 10.7717/peerj.453. [Online]. Available: https://doi.org/10.7717/peerj.453.

[9] T. Akiba, S. Sano, T. Yanase, and T. Ohta, *Optuna: A next-generation hyperparameter optimization framework*, https://optuna.org/, Accessed on May 14th, 2023.