Temporal Sequences of EEG Covariance Matrices for Automated Sleep Stage Scoring with Attention Mechanisms

Mathieu Seraphim[1](✉), Paul Dequidt[1], Alexis Lechervy[1],
Florian Yger[1,2], Luc Brun[1], and Olivier Etard[3]

[1] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
mathieu.seraphim@unicaen.fr
[2] LAMSADE, UMR CNRS 7243, Université Paris-Dauphine, PSL, Paris, France
[3] Université de Caen Normandie, INSERM, COMETE U1075, CYCERON,
CHU de Caen, Normandie Univ, 14000 Caen, France

Abstract. Electroencephalographic (EEG) data is commonly used in sleep medicine. It consists of a number of cerebral electrical signals measured from various brain locations, subdivided into segments that must be manually scored to reflect their sleep stage. These past few years, multiple implementations aimed at an automation of this scoring process have been attempted, with promising results, although they are not yet accurate enough with respect to each sleep stage to see clinical use. Our approach relies on the information contained within the covariations between multiple EEG signals. This is done through temporal sequences of covariance matrices, analyzed through attention mechanisms at both the intra- and inter-epoch levels. Evaluation performed on a standard dataset using an improved methodological framework show that our approach obtains balanced results over all classes, this balancing being characterized by a better MF1 score than the State of the Art.

Keywords: Sleep analysis · EEG · Deep Learning · Attention · Symmetric Positive Definite matrices

1 Introduction

To study sleep patterns in the field of sleep medicine, the gold standard is the polysomnography (PSG) study, which usually includes electroencephalography (EEG), electrooculography (EOG), electromyography (EMG) and electrocardiography (ECG) recordings, corresponding to brain, eye, muscle and heart electrical activity, respectively. These signals are derived from the voltage existing

This work has been co-funded by the Normandy Region and the French National Research Agency (ANR) through a HAISCoDe Ph.D. grant. It was granted access to the HPC resources of IDRIS under the allocation 2022-AD010613618 made by GENCI, and to the computing resources of CRIANN (Normandy, France).

N. Tsapatsoulis et al. (Eds.): CAIP 2023, LNCS 14185, pp. 67–76, 2023.
https://doi.org/10.1007/978-3-031-44240-7_7

between electrodes over time, often with one being set as a reference. In this paper, the term “signal” shall refer exclusively to such a voltage.

The set of norms most often used to analyze PSG signals is the one defined by the American Academy of Sleep Medicine (AASM) [4]. This analysis is done by subdividing the signals into 30 s epochs, sometimes called “sleep epochs” in this paper. These may be manually scored (labeled) as being in one of five stages: wakefulness, rapid eye movement (REM) sleep, and three stages of non-REM sleep (N1, N2 and N3).

Table 1. Frequency bands that we use for EEG data analysis

	Delta	Theta	Alpha	Beta$_{low}$	Beta$_{high}$	Gamma
Hz	[0.5, 4[	[4, 8[	[8, 12[	[12, 22[	[22, 30[	[30, 45[

In this paper, we study the relevance of cerebral functional connectivity as a tool for the automated classification of sleep stages, through a study of covariations between EEG signals. In particular, we aim to obtain a high level of class-wise performance. For that purpose, we analyze timeseries of covariance matrices, computed for various frequency bands (Table 1). We base our analysis on an existing model architecture [14], itself based on successive Transformer encoders. After an overview of the existing State of the Art (SOA) in Sect. 2, we shall explain our method in Sect. 3. Finally, in Sect. 4, we present our results on a commonly used dataset, including a comparison with SOA methods.

2 State of the Art

Some approaches consider that a single signal contains enough information to classify sleep epochs [12,14,21]. A common strategy is to combine an EEG and an EOG signal with the same reference electrode by subtracting them [15–17]. Other approaches use a multitude of input signals, often including EOG or EMG signals to said input, in addition to EEG. Phan et al. [11] use one signal of each type (EEG, EOG and EMG) as input, whereas Jia et al. [7,8] use multiple of each, and additionally include one ECG signal. Given the same dataset, the latter approaches seem to yield better results.

A common approach in EEG preprocessing pipelines is the extraction of relevant frequency components, since sleep stages are characterized by events with specific frequential components [4]. As such, Phan et al. [11,12,14] compute time-frequency images to use as input of their model.

Manual scoring of a sleep epoch takes into consideration said epoch’s context - i.e. information contained in neighboring sleep epochs. Similarly, the architectures of models used for this task often include contextual information in the classification process. Such sequence-based models can be divided into two sections: intra-epoch (extracting features from each epoch in the input sequence) and

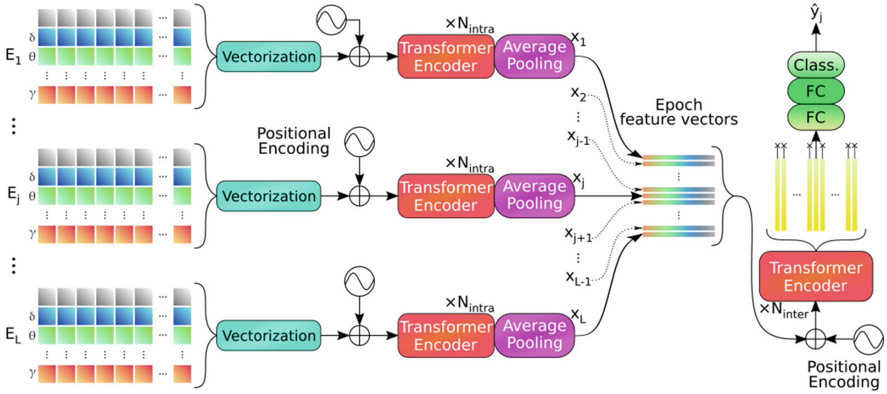

Fig. 1. Our model. $(E_1, ..., E_L)$ is the input sequence, with E_j referring to the central epoch. $\hat{y}_j$ is the output classification of the model. N_{intra} and N_{inter} refer to the number of sublayers in our intra- and inter-epoch Transformer encoders.

inter-epoch (combining said features). Convolutional neural networks (CNNs) can be used at the intra-epoch level, usually followed at the inter-epoch level by recurrent neural networks (RNNs) [12,15–17]. Phan et al. expand on both the RNN and attention mechanism approaches. In [11,12], they utilize bi-directional RNNs at both the intra-epoch and inter-epoch levels, whereas they use Transformer encoder-based attention mechanisms [18] in [14]. Similarly, Zhu et al. [21] use attention blocs inspired by said encoders at both levels, together with convolutions and other more classic layers. It has been stated that the performance of sequence-based State of the Art automatic sleep scoring models is currently near perfect, with little room for improvement [13]. While we do not dispute that claim in absolute terms, we have noticed a discrepancy in class-wise performance, particularly regarding the N1 stage (see Sect. 4.4). Therefore, our main focus is to correct for this discrepancy.

Our chosen axis of analysis concerns functional connectivity. In other words, one may study the connectivity between different brain regions through correlations detected between them, often independently of the structural (i.e. physical) connectivity between said regions [6]. In the context of sleep studies, it has been proven that sleep induces a characteristic cerebral response, describable in terms of functional connectivity [5]. Jia et al. [7,8] explicit these inter-region relationships through graph timeseries. Their intra-epoch section is a graph learning model, with each node corresponding to an electrode. These graphs are then convolved both spatially and temporally in the inter-epoch section. Note that most graph convolution methods do not assign a specific weight to each node, nor do they use the relative positioning of said nodes. For the proposed graphs, however, each node corresponds to an electrode, so ignoring node specificity in such a way might actually be a drawback.

In this paper, we perform an analysis of functional connectivity, estimated through the covariations of brain signals. For this purpose, we analyze covariance matrices computed from multiple simultaneous EEG signals, excluding other signal types (EOG, EMG...) in order to focus exclusively on brain activity. Covariance matrices are guaranteed to be symmetric positive semi-definite, but tend to be fully symmetric positive definite (SPD) when computed from real-world data. The set of all SPD matrices in $\mathbb{R}^{n \times n}$ is a Riemannian manifold (metered curved space), and we postulate that preserving this geometry in our model would be advantageous to our classification, as similar approaches using SPD matrices have already been implemented in the field of EEG signal analysis, most notably in brain-computer interfaces (BCI) [19].

3 Method

3.1 From EEG Signals to Covariance-Derived SPD Matrices

As do Zhu et al. [21], we apply a z-score normalization to our EEG signals, in order to harmonize their means and standard deviations. Moreover, according to the AASM [4], the signal components indicative of the current sleep stage have specific frequential properties. In order to allow the network to more effectively analyze them, we filter our EEG signals along the six frequency bands presented in Table 1. This is done through a fourth-order Butterworth bandpass filter.

The discrete events indicative of a sleep epoch's proper classification are around one second in length. To capture them, we elected to subdivide our recordings into one second segments. Each sleep epoch is therefore subdivided into 30 non-overlapping segments. On each segment, we compute a covariance matrix between the n electrodes. We verify that the resulting matrices are properly SPD, and add the matrix $\mathbb{I}_n \times 10^{-5}$ to those who aren't. This is done on the unfiltered and filtered signals, resulting in a total of 7 data channels.

Two main families of metrics have been defined on the set of SPD matrices. The so-called affine invariant metrics [10] are invariant to affine transformations, but have some drawbacks - for instance, it is impossible to compute an algebraic mean using such a metric, though algorithmic approximations do exist. LogEuclidean metrics [2] do not showcase the same invariance properties, but are significantly easier to work with. The LogEuclidean distance between two SPD matrices A and B is defined as:

$$\delta_{LE}^{P}(A, B) = \|log(P^{-1/2}AP^{-1/2}) - log(P^{-1/2}BP^{-1/2})\|_F \tag{1}$$

This metric relies on the bijection existing between the manifold and its tangent space, the space of symmetric matrices, by way of the matrix logarithm and exponential functions. The parameter P may be interpreted as a center of projection onto said space.

Given a covariance matrix, the only mono-signal information stored is the variance of the signal along the segment. Additional signal-specific features may be added using Eq. 2, which "augments" a covariance matrix C, preserving its SPD property while adding a feature vector V (referred to as a "side vector"), weighted by a factor α (with $V_\alpha = \alpha V$):

$$M = \left(\begin{array}{c|c} C + V_\alpha V_\alpha^T & V_\alpha \\ \hline V_\alpha^T & 1 \end{array}\right) \tag{2}$$

Each epoch entering the model is thus represented by 7 channels of 30 SPD covariance matrices, and their associated side vectors. Multiple side vectors may be computed per matrix, such as its mean, maximum value, or average power spectral density (PSD) over the corresponding one second segment.

Being biological, our EEG data is marked by the specificities inherent to each recording, that are then transferred to our covariance matrices. In order to reduce said specificities, we compute every recording-wise covariance matrix G, and use them to apply a whitening operation [20] onto the relevant matrices:

$$M' = G^{-1/2} M G^{-1/2} \tag{3}$$

The idea is to operate a "transport" of the data M centered around G to be centered around $\mathbb{I}_n$ instead. We perform this shift for each recording and compute distances between centered SPD matrices using Eq. 1, with $P = \mathbb{I}_n$. If need be, both M and G are augmented with the relevant side vectors.

3.2 The Model

Our model architecture uses Transformer encoders at the intra- and inter-epoch levels, as does [14]. It takes as input a timeseries of sleep epochs, composed of a central epoch and l epochs on either side, for a total of $L = 2l+1$. These sequences are constructed with maximum overlap, with classification on the central epoch. Thus, the first and last l epochs of each recording are not classified.

Our model starts with a vectorization layer. It performs the augmentation of matrices by their weighted side vectors (Eq. 2), followed by the whitening operation. The nature of the side vectors V, and the value of their weight α, are model hyperparameters. Using n electrodes, we project our SPD matrices of $\mathbb{R}^{(n+1)\times(n+1)}$ onto their tangent set (Sect. 3.1), and vectorize the upper triangular of the resulting symmetric matrix onto $\mathbb{R}^{\frac{(n+1)(n+2)}{2}}$ [2]. These operations being bijective, all Euclidean operations on these vectors are interpretable as LogEuclidean operations on the augmented matrices.

These vectors undergo a positional encoding [18]. The channels are then concatenated and fed to a first, intra-epoch Transformer encoder, composed of a number of sequential sublayers. The fully connected layers present in each encoder sublayer allow for a mixing of the elements of each input vector, and therefore a mixing of the original channels. In order to obtain a single feature vector per sleep epoch, the output of the intra-epoch encoder layer passes through an average pooling layer. The resulting L epoch feature vectors are then fed through another positional encoding layer, followed by an inter-epoch encoder.

Only the output vector corresponding to the central sleep epoch is preserved, passing through two fully connected layers, each followed by a ReLU activation and a dropout layer. A final fully connected "classification" layer reduces the output to the desired 5 data points (one per class), and this classification is then fed to a softmax-including cross-entropy loss function.

We optimize this model using the Adam algorithm, with the function parameters β_1, β_2 and ϵ set to 0.9, 0.999 and 10^{-7} respectively. The weight decay is a hyperparameter, and so are the model's learning rate λ and the corresponding exponential decay parameter γ_λ.

Our architecture can be seen in Fig. 1. The number of sublayers and attention heads of each encoder, the size of parameter tensors for the fully connected layers and the various dropout probabilities are all hyperparameters. Our hyperparameter-obtaining strategy is described in Sect. 4.2 , and the obtained values are presented in the annex.

4 Experiments

4.1 Dataset Used

We chose to validate our model on the SS3 subset of the Montreal Archive of Sleep Studies (MASS) dataset [9], as it is heavily utilized within the SOA and contains a large number of electrodes to choose from for our analysis. Said subset is made up of 62 subjects, with a single full-night recording per subject and 20 EEG channels in common. Each EEG signal went through a notch filter at 60 Hz as well as a lowpass and highpass filter with cutoff frequencies of 0.30 Hz and 100 Hz respectively. This dataset is unbalanced, with the largest and smallest classes (N2 and N1) respectively containing 50.24% and 8.16% of its sleep epochs.

In order to capture a significant range of signals, and to limit redundancy between neighboring electrodes, we chose electrodes F3, F4, C3, C4, T3, T4, O1 and O2. This selection has a relatively homogeneous distribution with regards to the cranium, with inter-hemispheric symmetry to capture relevant variations along that axis. All of these signals are captured with a common reference electrode, located behind the left ear.

4.2 Model Validation

As is best practice, we subdivide our database into three subsets: training, validation and test. We utilize a k-fold cross-validation scheme, using the same fold-wise subset separation as Seo et al. [15] in order to facilitate comparisons. Each of the $k = 31$ folds are divided into 50, 10 and 2 recordings for each training, validation and testing set respectively. The 31 folds' testing sets add up to the 62 recordings in SS3, with no overlap. We set the parameter l of our network to 10, as is done in [14]. We rebalance each fold's training set through oversampling, with each class having as many elements as N2 has. The validation and test sets aren't rebalanced, though test sets are further restricted (Sect. 4.3).

Every hyperparameter research is ran using the Tree-structured Parzen Estimator algorithm [3], as implemented by Optuna [1]. This research is done on the same randomly selected fold. The best hyperparameters are then utilized to train the model on all folds. We use the macro-averaged F1 score (MF1) as our main performance statistic, as it reflects imbalances in class-wise classification performance, and is widely used throughout the SOA. All statistics are summarized over the 31 folds by computing their mean and standard deviation.

Table 2. Ablation study and comparison to the SOA.

		Balanced statistics		Unbalanced statistics	
	Method	MF1	Macro accuracy	General accuracy	Kappa
0	SleepTrans. [14]	73.97 ± 3.50	76.37 ± 4.35	81.25 ± 3.54	0.722 ± 0.046
1	IITNet [15]	78.48 ± 3.15	**81.88** ± 2.89	83.90 ± 3.03	0.763 ± 0.043
2	DeepSleepNet [16]	78.14 ± 4.12	80.05 ± 3.47	84.81 ± 3.70	0.773 ± 0.052
3	GraphSleepNet [8]	75.58 ± 3.75	79.75 ± 3.41	80.97 ± 4.35	0.724 ± 0.057
4	Our method	**79.78** ± 4.56	81.76 ± 4.61	**85.05** ± 4.97	**0.776** ± 0.069
5	No covariance	77.39 ± 5.82	79.76 ± 4.95	82.61 ± 6.01	0.741 ± 0.081
6	No side vectors	78.14 ± 4.10	80.56 ± 3.95	83.38 ± 4.16	0.753 ± 0.060

Table 3. F1 scores per class.

	Method	N3 F1	N2 F1	N1 F1	REM F1	Wake F1
0	[14]	74.26 ± 12.36	86.72 ± 3.28	47.60 ± 6.37	83.84 ± 6.99	77.40 ± 8.63
1	[15]	**81.97** ± 8.91	88.15 ± 2.84	56.01 ± 6.54	85.14 ± 5.64	81.11 ± 8.49
2	[16]	80.38 ± 9.35	**89.25** ± 3.12	53.52 ± 8.24	86.67 ± 5.34	80.86 ± 9.04
3	[8]	74.77 ± 12.12	84.84 ± 4.22	50.80 ± 8.06	85.09 ± 7.38	82.42 ± 7.43
4	Ours	78.17 ± 11.49	88.66 ± 4.59	**58.43** ± 6.41	**86.91** ± 7.79	**86.73** ± 6.42

4.3 Reproducing the State of the Art

In order to compare our results to the State of the Art, we selected four approaches. hree of those are DeepSleepNet [16], often used as a benchmark, IITNet [15], whose cross-validation folds we are using, and GraphSleepNet [8], which also analyses functional connectivity. The fourth, SleepTransformer [14], shall be discussed subsequently.

All three have their code available on GitHub, and were trained on MASS SS3 in their respective papers. IITNet, GraphSleepNet and DeepSleepNet use sequences of epochs as inputs, of size equal to 10, 5 and 25 respectively. Like us (Sect. 3.2), IITNet and GraphSleepNet use each sequence to classify a single epoch, respectively the last and central epoch of the sequence. In contrast, DeepSleepNet outputs one classification per epoch in their sequences, which are constructed without overlap. Because of this, for each recording, IITNet won't classify the first 9 epochs, GraphSleepNet will ignore the first and last 2, and DeepSleepNet might ignore up to 24 epochs at the end.

All three models use a similar results aggregation strategy. For each fold, the best trained parameters are used to compute predictions on the test set. Despite originating from different models, these predictions are concatenated, and statistics are computed over this unified predictions tensor. As the number of sleep epochs per recording is not homogeneous, neither are the test sets. This strategy therefore results in an implicit weighting effect, giving more importance to sets of parameters computed on folds with larger test sets.

In order to better compare these methods to our model, we retrained these models with our metrics, folds, and results summarizing methods (Sect. 4.2). All methods were adapted to select their best fold learned parameters through their validation MF1 score. In the spirit of fairness, we rebalanced GraphSleepNet and IITNet's training sets through oversampling. DeepSleepNet already does this when pretraining its intra-epoch submodel, and its multi-label sequences can't be rebalanced in that way. We did not change any of their model architectures, and used their published hyperparameters.

The fourth SOA method presented is our reimplementation of the original SleepTransformer model. Compared to our model, this method uses a custom attention softmax layer instead of our average pooling. We also replicated their preprocessing using a recombined Fz-Cz signal from MASS SS3. It was trained with our methodology, including a hyperparameter research.

The obtained results (Tables 2 and 3) differ from those originally published, which may stem for the aforementioned methodological differences. To harmonize all test sets, we have elected to exclude the classification of the first and last 24 epochs of each recording. The training or validation sets remain, however, unchanged. This has been applied to all results presented in this paper.

4.4 Analysis of Results

Aside from lines 1, 2 and 3 of Tables 2 and 3, all presented results are preceded by a hyperparameter research.

Line 0 of Tables 2 and 3 show us the results obtained through our reimplementation of SleepTransformer. As we can see, they are the lowest of all presented methods. Due to the similarities between our approaches, one might view these as the baseline for our architecture's performance.

As stated in Sect. 3.2, we tested multiple side vector types in our hyperparameter research. The one that consistently performed the best was the vector of mean PSDs. The other chosen hyperparameters are described in the annex.

The last 3 lines of Table 2 give an overview of the obtained results. Line 4 corresponds to our results, trained on the best hyperparameters mentioned above. A surprising hyperparameter is the value of α (Sect. 3.1) of 99.53. This implies that the side vectors have a large impact on the final classification, and thus that our network favors a signal-specific input (one not obtained through covariance). To assess the relevance of covariances altogether, we removed all covariance information from our data (i.e. the non-diagonal elements of the covariance matrices), and reran our model. As seen in line 5 of Table 2, all statistics but Kappa are lower than the ones of line 4 by about 2%. This is coherent with the literature, as decent performances have been obtained on MASS without relying on covariations. We also trained our model on the original covariance matrices themselves, with no side vector augmentation (as seen in line 6). We obtain similar results to line 4 and superior results to line 5, thus implying that considering covariations adds a net benefit.

When it comes to the rest of the reran State of the Art, lines 1 through 4 of Table 2 shows that our model performs better in all measured metrics except for macro-averaged accuracy, where we are a close second. In addition, Table 3 shows that our method outperforms the others in REM, Wake and N1 sleep classification. As seen by the scores and standard deviations, though, the quality of predictions varies much per class, for both the State of the Art and us. In particular, N1 sleep epochs seem particularly hard to classify, but our method shows a two points lead over the next best one in that regard. This lead would explain our ranking in terms of MF1 score (Table 2).

All-in-all, Tables 2 and 3 show that a method based in part on covariance information provides results either equivalent or superior to the State of the Art on this problem (relative to the chosen statistics), with notable improvements to performance on the N1 stage, though it also benefits from signal-specific inputs.

5 Conclusion

We have presented our novel approach for automatic scoring of sleep stages through an analysis of the covariations between EEG signals. Motivated by the high imbalance between the classification of said stages, we established a fairer methodology for training and validating models on this problem. The results validate our hypothesis on the relevance of such covariations in this context, and by extension, that of functional connectivity.

Appendix

The hyperparameters corresponding to the best version of our model are:
Side vectors: PSD; α: 99.53; intra-epoch encoder: 5 sublayers, 15 attention heads, fully connected components of size 1024, dropout of 6.2×10^{-5}; intra-epoch encoder: 6 sublayers, 5 attention heads, fully connected components of size 256, dropout of 8.1×10^{-3}; final fully connected layers: of size 2048, dropout of 1.4×10^{-3}; learning rate (λ): 4.9×10^{-5}, γ_λ at 0.94; weight decay at 1.76×10^{-6}.

Many thanks to Huy Phan [11–14] for answering all our questions.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631 (2019)

2. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-euclidean metrics for fast and simple calculus on diffusion tensors. Magn. Reson. Med. **56**(2), 411–421 (2006)
3. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Advances in Neural Information Processing Systems, vol. 24 (2011)
4. Berry, R.B., et al.: AASM scoring manual updates for 2017 (version 2.4) (2017)
5. Bouchard, M., Lina, J.M., Gaudreault, P.O., Dubé, J., Gosselin, N., Carrier, J.: EEG connectivity across sleep cycles and age. Sleep **43**(3) (2019)
6. Eickhoff, S., Müller, V.: Functional connectivity. In: Toga, A.W. (ed.) Brain Mapping, pp. 187–201. Academic Press, Waltham (2015)
7. Jia, Z., et al.: Multi-view spatial-temporal graph convolutional networks with domain generalization for sleep stage classification. IEEE Trans. Neural Syst. Rehabil. Eng. **29**, 1977–1986 (2021)
8. Jia, Z., et al.: Graphsleepnet: adaptive spatial-temporal graph convolutional networks for sleep stage classification. In: IJCAI, pp. 1324–1330 (2020)
9. O'reilly, C., Gosselin, N., Carrier, J., Nielsen, T.: Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research. J. Sleep Res. **23**(6), 628–635 (2014)
10. Pennec, X., Fillard, P., Ayache, N.: A riemannian framework for tensor computing. Int. J. Comput. Vision **66**(1), 41–66 (2006)
11. Phan, H., Chén, O.Y., Tran, M.C., Koch, P., Mertins, A., De Vos, M.: Xsleepnet: multi-view sequential model for automatic sleep staging. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 5903–5915 (2022)
12. Phan, H., et al.: L-seqsleepnet: whole-cycle long sequence modelling for automatic sleep staging (2023)
13. Phan, H., Mikkelsen, K.: Automatic sleep staging of EEG signals: recent development, challenges, and future directions. Physiol. Meas. **43**(4), 04TR01 (2022). https://doi.org/10.1088/1361-6579/ac6049
14. Phan, H., Mikkelsen, K., Chén, O.Y., Koch, P., Mertins, A., De Vos, M.: Sleeptransformer: automatic sleep staging with interpretability and uncertainty quantification. IEEE Trans. Biomed. Eng. **69**(8), 2456–2467 (2022)
15. Seo, H., Back, S., Lee, S., Park, D., Kim, T., Lee, K.: Intra- and inter-epoch temporal context network (IITNET) using sub-epoch features for automatic sleep scoring on raw single-channel eeg. Biomed. Signal Process. Control **61**, 102037 (2020)
16. Supratak, A., Dong, H., Wu, C., Guo, Y.: Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg. IEEE Trans. Neural Syst. Rehabil. Eng. **25**(11), 1998–2008 (2017)
17. Supratak, A., Guo, Y.: Tinysleepnet: an efficient deep learning model for sleep stage scoring based on raw single-channel EEG. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), pp. 641–644 (2020)
18. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
19. Yger, F., Berar, M., Lotte, F.: Riemannian approaches in brain-computer interfaces: a review. IEEE Trans. Neural Syst. Rehabil. Eng. **25**(10), 1753–1762 (2017)
20. Yger, F., Sugiyama, M.: Supervised logeuclidean metric learning for symmetric positive definite matrices (2015)
21. Zhu, T., Luo, W., Yu, F.: Convolution-and attention-based neural network for automated sleep stage classification. Int. J. Environ. Res. Public Health **17**(11), 4152 (2020)